Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

1992-12

Modeling and classification of biological signals

VanDerKamp, Martha M.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/23965

Downloaded from NPS Archive: Calhoun

http://www.nps.edu/library

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) EC	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a NAME OF FUNDING / SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
MODELING AND CLASSIFICATION OF BIOLOGICAL SIGNALS

12 PERSONAL AUTHOR(S)
VANDERKAMP, Martha M.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	1992 December	97

16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the US Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Autoregressive Modeling; Biological Classification; Neural Networks; Itakura Distance Measure

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This thesis examines a number of marine biological signals and the problem of modeling by autoregressive techniques using a prony-svd algorithm to accurately represent segments of biological signals. Two methods are employed to classify the biological signals from the model parameters. The first classification method is based on a Neural Network implementation using a commercial software package. The second method is accomplished by using a distance measure, based on spectral ratios, with respect to modeled reference signals.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL CRISTI, Roberto	22b TELEPHONE (Include Area Code) 408-646-2223 22c OFFICE SYMBOL EC/Cx

DD Form 1473, JUN 86 *Previous editions are obsolete.*

S/N 0102-LF-014-6603

T259117

Approved for public release; distribution is unlimited.

Modeling and Classification of Biological Signals

by

Martha M. VanDerKamp
Lieutenant, United States Navy
B.S., University of the State of New York, 1983

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

ABSTRACT

This thesis examines a number of marine biological signals and the problem of modeling by autoregressive techniques using a prony-svd algorithm to accurately represent segments of biological signals. Two methods are employed to classify the biological signals from the model parameters. The first classification method is based on a Neural Network implementation using a commercial software package. The second method is accomplished by using a distance measure, based on spectral ratios, with respect to modeled reference signals.

176132
C.1

TABLE OF CONTENTS

I. INTRODUCTION 1

II. PARAMETRIC MODELING OF BIOLOGICAL SIGNALS 4

A. PRONY'S METHOD 5

B. SINGULAR VALUE DECOMPOSITION (SVD) 10

C. MODEL ORDER SELECTION 13

III. MODEL DEVELOPMENT OF BIOLOGICAL SIGNALS 19

A. TIME DOMAIN SIGNAL SOURCE 19

B. PRONY-SVD METHOD DETAILED DESCRIPTION 28

C. SOFTWARE IMPLEMENTATION OF THE PRONY-SVD METHOD 30

IV. CLASSIFICATION METHODS 33

A. NEURAL NETWORKS 33

1. Neural Network Description and Applicability 33

2. Back-Propagation Network 37

a. Delta Rule Learning Law 41

b. Transfer Function 42

c. Back-Propagation Network Summary . . . 43

B. ITAKURA DISTANCE CLASSIFICATION 44

1. Symmetrized Itakura Distance Measure . . . 45
a. SI Distance Measures Applied to AR Processes 46
2. Implementing the Symmetrized Itakura Distance 50
3. Data Manipulation 51

V. CLASSIFICATION RESULTS 53
A. TRAINING THE NEURAL NETWORK 53
B. PERFORMANCE OF THE NEURAL NETWORK 54
C. PERFORMANCE OF THE SYMMETRIZED ITAKURA DISTANCE MEASURE CLASSIFICATION 59

VI. SUMMARY, CONCLUSION, AND RECOMMENDATIONS 69
A. SUMMARY AND CONCLUSION 69
B. RECOMMENDATIONS 71

APPENDIX A - SOFTWARE AND HARDWARE DESCRIPTION 73

APPENDIX B - COMPUTER CODE 75
A. ITAKURA.M 75
B. ITAREF.M 77
C. PRONYSVD.M 79
D. RESULT3.M 80
E. RESULTS.M 81
F. RMSE.M 81

G. RUNDATA.M 82

APPENDIX C - NEURAL NETWORK SCORING PERCENTAGES 83

REFERENCES 87

INITIAL DISTRIBUTION LIST 88

ACKNOWLEDGEMENTS

I would like to acknowledge the cooperation of Nicole Crane of the Hopkins Marine Station of Stanford University who provided the majority of data for this thesis.

No acknowledgment would be complete without recognizing those people who make the thesis possible. I wish to especially thank my Thesis Advisor, Professor Roberto Cristi and my Co-Advisor, Professor Monique Fargues, for their superlative assistance in this undertaking. Also I want to thank my son Cameron who persevered with loving patience through it all.

I. INTRODUCTION

The objective of this thesis is to develop and test autoregressive (AR) modeling techniques to a number of transient signals of biological origin. Once suitably modeled, the feasibility of classification of these biological signals from the modeled parameters will be explored. Two methods of classifications will be attempted. One method uses neural networks while the other method is based on a spectral ratio distance measurement.

Traditional signal processing techniques based on Fourier Transforms are not well suited to capture features of short signals, such as transients due to their nonstationary nature. Parametric modeling techniques such as AR modeling, yield better results since they exploit the fact that sound signals are generated by a few dominant frequency components. The ability to apply a least squares technique to estimate the AR model parameters makes it attractive from an implementation standpoint.

The advantage of using parametric models are twofold: a) the dominant frequencies together with the spectrum of the signal can be estimated from relatively short duration data, and b) the low order parametrization can be used in a pattern recognition and classification framework.

In this thesis, we make use of the parametric modeling of the biological data to develop an automatic unsupervised classification scheme. Two basic approaches will be considered and compared: a neural network and a classifier based on spectral ratio.

The strength of a neural network to be used as a biological classifier lies in the networks nonlinear characteristic and its ability to be trained to recognize many types of data structures. The nonlinearity of the network allows for an effective association between data having similar structures. This similarity matching allows a network trained to recognize a set of biological data to make classifications on the biologics based on similarities taught to the system simply by training. In this way, the output of the neural network provides a real-time biological classification.

The Symmetrized Itakura (SI) distance is a spectral distance measure that will be utilized as the second classification technique. The properties of the SI distance make it a possible choice among spectral distances in classification of biological signals. The SI distance is symmetric, that is the distance $d(x_1,x_2)$ between two transient signals x_1 and x_2 is the same as the distance $d(x_2,x_1)$ between x_2 and x_1. But most importantly, the SI distance is independent of the magnitude of the transient sound. Both

classification methods are implemented based on the AR model parameters.

The thesis will conclude by comparing the methods of classification. The primary source of data for this thesis was provided by the Hopkins Marine Station of Stanford University located in Pacific Grove, California.

II. PARAMETRIC MODELING OF BIOLOGICAL SIGNALS

The biological signals available for classification consist of data records in excess of 300,000 bytes. The length of the data precludes a direct use for classification. The amount of data can be drastically reduced by developing parameter methods to represent the signal characteristics. The main features of acoustic signals (in air or water) can be characterized in the frequency domain. This comes from the very way they are generated, as resonating cavities and vibrations of vocal chords. As a consequence most of the energy is concentrated on a few frequency components.

It is this feature that is the basis of an effective parametrization of sound signals. In this chapter we review the concept of modeling using Prony's method, and we present techniques to identify the main frequency components present in the signals of interest. This leads to parametric methods, where the information contained in relatively long data sets (512 or 1024 points) can be summarized in a much smaller number of parameters (in our case about 20). These parameters are then used for classification, as we will see in subsequent chapters.

A. PRONY'S METHOD

Prony's method is a technique for modeling sampled data as a linear combination of exponentials. It can be used in the context of spectral estimation and leads to linear prediction and autoregressive (AR) modeling. Prony's method models the data as a superposition of complex exponentials or damped sinusoidal models. The parameters of the exponentials are determined by a least squares fit to the data.

Given N complex data samples $x[1],\ldots,x[N]$, the Prony method models $x[n]$ with a p-term complex exponential model

$$\hat{x}[n] = \sum_{k=1}^{p} A_k \exp\left[(\alpha_k + j2\pi f_k)(n-1)T + j\theta_k\right], \qquad (2.1)$$

for $1 \le n \le N$, where T is the sample interval in seconds, A_k is the amplitude of the complex exponential, α_k is the damping factor in seconds^{-1}, f_k is the sinusoidal frequency in Hz, and θ_k is the sinusoidal initial phase in radians. In the case of real data samples, the complex exponentials must occur in complex conjugate pairs of equal amplitude, thus reducing the exponential representation to

$$\hat{x}[n] = \sum_{k=1}^{\frac{p}{2}} 2A_k \exp\left[\alpha_k (n-1)T\right] \cos\left[2\pi f_k (n-1)T + \theta_k\right], \qquad (2.2)$$

for $1 \leq n \leq N$. When the number of complex exponentials p is even, there are $p/2$ damped cosines, and when the number is odd there are $(p-1)/2$ damped cosines plus a single purely damped exponential. [Ref. 1:pp. 303-304]

Generally, $x[n]$ is observed in noise,

$$y[n] = \hat{x}[n] + \epsilon[n], \tag{2.3}$$

where $y[n]$ is the observed data and $\epsilon[n]$ is a white noise process.

The biological data consists of a number of data points N much greater than the minimum number needed to fit a model of p exponentials, i.e., $N >> 2p$. This requires the parameters to be estimated by a least squares fit.

In order to determine the parameters of the model, let us write Equation (2.1) as the result of an overdetermined case and the data sequence as

$$\hat{x}[n] = \sum_{k=1}^{p} h_k z_k^{n-1}, \tag{2.4}$$

where $h_k = A_k \exp(j\theta_k)$ and $z_k = [(\alpha_k + j2\pi f_k)$ for $1 \leq n \leq N$ [Ref. 1:p. 306]. The error is denoted as $\epsilon[n] = x[n] + \hat{x}[n]$, where $\hat{x}[n]$ is the estimated signal. The total squared error is minimized by simultaneously finding the order p and the parameters $\{h_k, z_k\}$ for $k = 1$ to $k = p$. Although the minimization of the squared error

$$\rho = \sum_{n=1}^{N} |\varepsilon[n]|^2 \qquad (2.5)$$

is a nonlinear problem it can be solved by standard techniques using the covariance linear prediction normal equations as encountered for AR spectral analysis.

The estimation of the parameters z_k can be carried out by observing that $\hat{x}[n]$ is the solution of a difference equation of the form

$$x[n] + a_1 x[n-1] + \ldots + a_p x[n-p] = 0, \qquad (2.6)$$

with 1, $a_1, \ldots a_p$ the coefficients of the polynomial $A(z) = (1 - z_1 z^{-1})(1 - z_2 z^{-1}) \ldots (1 - z_p z^{-1})$. From this, the connection to a linear prediction model where

$$x[n] = -a_1 x[n-1] - \ldots - a_p x[n-p] \qquad (2.7)$$

is straightforward. The parameters a_i can then be determined by solving a set of linear equations

$$\begin{Vmatrix} c_{xx}[1,1] & c_{xx}[1,2] & \cdots & c_{xx}[1,p] \\ c_{xx}[2,1] & c_{xx}[2,2] & \cdots & c_{xx}[2,p] \\ \vdots & \vdots & \ddots & \vdots \\ c_{xx}[p,1] & c_{xx}[p,2] & \cdots & c_{xx}[p,p] \end{Vmatrix} \begin{Vmatrix} \hat{a}[1] \\ \hat{a}[2] \\ \vdots \\ \hat{a}[p] \end{Vmatrix} = - \begin{Vmatrix} c_{xx}[1,0] \\ c_{xx}[2,0] \\ \vdots \\ c_{xx}[p,0] \end{Vmatrix}, \qquad (2.8)$$

where

$$c_{xx}[j,k] = \frac{1}{N-p}\sum_{n=p}^{N-1} x^*[n-j]\,x[n-k]\,. \qquad (2.9)$$

The resulting minimum modeling error is computed as

$$\hat{\sigma}^2 = \hat{\rho}_{MIN} = c_{xx}[0,0] + \sum_{k=1}^{p} \hat{a}[k]\,c_{xx}[0,k]\,. \qquad (2.10)$$

The matrix in Equation (2.8) is hermitian and positive semidefinite. It is singular if the data consist of p - 1 or fewer complex sinusoids. Any noise in the observation will cause the matrix to be nonsingular. It turns out that the solution of Equation (2.8) yields a stable model with poles inside the unit circle. From the above matrix, it is seen that $c_{xx}[j,k]$ is an estimate of $r_{xx}[j - k]$, but a different estimate from the autocorrelation method. The matrix of Equation (2.8) uses the sum of only $N - p$ lag products to estimate the autocorrelation function for each lag even though more data is available. As an example, in the estimation of $r_{xx}[0]$ the biased autocorrelation estimator of the autocorrelation method uses all N data points, while the covariance method uses only $N - p$ data points in the summation. When dealing with very large data records, $N >> p$, the "end effects" due to the missing p points are negligible and as a result, the autocorrelation and covariance methods

produce similar results. A second contrasting feature is that for data consisting of pure sinusoids, the covariance method may be used to extract the frequencies. This is due to the similarities between the Prony and covariance methods. Note that this property is not shared by the autocorrelation method. [Ref. 2:p. 223]

The covariance method is identical to the modern version of Prony's method for pole estimation. From the estimated parameters $a_1, \ldots, a_p$, the frequencies z_k can be easily estimated as roots of the polynomial

$$A(z) = 1 + a_1 z^{-1} + \cdots + a_p z^{-p}. \qquad (2.11)$$

Furthermore, an estimate of the spectrum of the signal $x(n)$ is determined from the recursive AR model

$$x(n) + a_1 x(n-1) + \cdots + a_p x(n-p) = e(n), \qquad (2.12)$$

with $e(n)$ denoting the modeling error. Ideally $e(n)$ is a white gaussian sequence with variance σ^2, which leads to the frequency spectrum of $x(n)$ given by:

$$\mathbf{X}(e^{j\theta}) = \frac{\sigma^2}{|A(e^{j\theta})|^2}, \qquad (2.13)$$

where $0 \leq \theta \leq 2\pi$. This frequency spectrum should match the dominant components of the spectrum of the data $x(n)$.

When noise is present the original Prony method performs poorly since the recursive difference equation no longer holds [Ref. 2:p. 224]. The extended Prony method was developed to

handle the cases of exponential signals in noise. The derivation of the extended Prony will not be discussed here. In this thesis, the application of a singular value decomposition (SVD) approach to the Prony method is used instead to increase robustness in the presence of measurement noise.

B. SINGULAR VALUE DECOMPOSITION (SVD)

Singular value decomposition (SVD) is utilized with the Prony method to provide further improvement in the estimation of the model components. The SVD approach can be viewed as a nonlinear filtering technique.

The SVD technique takes an arbitrary m x n complex-valued matrix A of rank k and decomposes the matrix in terms of a m x m unitary matrix $U = [u_1 \ldots u_m]$, a n x n unitary matrix $V = [v_1 \ldots v_n]$, and a set of positive real numbers $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k > 0$ (singular values of A) so that the matrix A can be expressed as

$$A = U\Sigma V^H = \sum_{i=1}^{k} \sigma_i u_i v_i^H, \qquad \textbf{(2.14)}$$

where the m x n matrix Σ has the structure

$$\Sigma = \begin{Vmatrix} D & 0 \\ 0 & 0 \end{Vmatrix}, \qquad (2.15)$$

and D = diag($\sigma_1, \ldots \sigma_k$), is a k x k diagonal matrix. We call σ_i the singular values of the matrix A. Since U^HU = I, and V^HV = I, then

$$A^HA = V(\Sigma^H\Sigma)\,V^H \qquad A^HA = U(\Sigma\Sigma^H)\,U^H, \qquad (2.16)$$

and

$$A^HAv_i = \sigma_i^2 v_i \qquad AA^Hu_i = \sigma_i^2 u_i \qquad (2.17)$$

for $1 \le i \le k$. The matrix products $\Sigma^H\Sigma$ and $\Sigma\Sigma^H$ are diagonal matrices of size n x n and m x m, respectively, with diagonal elements σ^2_i. The matrices A^HA and AA^H are Hermitian of size n x n and m x m respectively. It can be easily seen that the columns of U are the orthonormal eigenvectors of AA^H and the columns of V are the orthonormal eigenvectors of A^HA [Ref. 1:p. 77]. Both U and V share the same eigenvalues σ^2_i for $1 \le i \le k$. Utilizing the unitary properties of U and V, the following is obtained

$$AV = U\Sigma V^HV = U\Sigma \qquad U^HA = U^HU\Sigma V^H = \Sigma V^H, \qquad (2.18)$$

or

$$Av_i = \sigma_i u_i \qquad A^Hu_i = \sigma_i v_i, \qquad (2.19)$$

for $1 \leq i \leq k$. The last equations show a relationship between the eigenvectors u_i and v_i corresponding to the same singular values σ_i.

The pseudoinverse A^* of the m x n matrix A of rank k is defined in terms of the SVD matrix components as the unique matrix

$$A^{\bigstar} = V\Sigma^{\bigstar}U^H = \sum_{i=1}^{k} \sigma_i^{-1} v_i u_i^H, \tag{2.20}$$

where

$$\Sigma^{\bigstar} = \begin{Vmatrix} D^{-1} & 0 \\ 0 & 0 \end{Vmatrix}. \tag{2.21}$$

The pseudoinverse of A^* provides the minimum-norm least squares solution $\underline{x} = A^*\underline{b}$ to the system of equations $A\underline{x} = \underline{b}$. When $m = n$ and the rank of A is n, the pseudoinverse of A is the same as the square matrix inverse, $A^* = A^{-1}$. When $m > n$ and the rank of A is n, then $A^* = (A^HA)^{-1}A^H$. The least squares solution for $\underline{x}$ is $\underline{x} = (A^HA)^{-1}A^H\underline{b}$, for a set of overdetermined equations. When $n > m$ and the rank of A is m, then $A^* = A^H(AA^H)^{-1}$ and $\underline{x} = A^*\underline{b}$. This is the minimum norm solution for a set of underdetermined equations due to the pseudoinverse resulting in the vector $\underline{x}$ of minimum norm. The pseudoinverse of A^* uses the SVD to determine the rank of A by examining the

number of non-negligible singular values. This manner is preferred over computing A^* directly from either $(A^HA)^{-1}A^H$ or $A^H(AA^H)^{-1}$. [Ref. 1:pp. 76-77]

The use of the SVD technique to reduce noise components in the modeling equations and to aid in discriminating the estimated model signal from noise components will be further discussed in Chapter III.

C. MODEL ORDER SELECTION

The best choice of the AR model order is not usually known ahead of time so it necessary to conjecture several model orders. An error criteria is established to determine the "best" model order to choose. If the order chosen is too low, some spectral components are not estimated, while too high an order introduces extra components not present in the original signal. Thus, model order selection is a tradeoff between increased resolution and decreased variance in the estimated spectrum. The most important aspect of model order selection is the quality of the spectral estimation. Although many model order estimators are available, few guidelines as to their use in practical applications are available [Ref. 2:p. 237]. For good spectral resolution with few spurious peaks, the AR model order should be chosen such that $N/3 < p < N/2$. Where N is the number of data samples to be modeled.

The choice of the biological signals model order was made by inspection rather than by using a statistical method. It

was decided to break down the biological signals into data segments of 512 samples. Using the above numerical guidelines, this would have resulted in a model order between 170 to 256. This is not a practical solution for order selection when considering the application of the model for classification methods. Through trial and error, the biological signals where run through the modelling algorithm and the fast Fourier transform (FFT) of the model was compared to that of the original signal. From the modelling algorithm, it can be seen that the singular values from the covariance matrix go quickly to zero. Figures 1 through 4 show examples of these singular values from four biological signals composed of 512 segments, with model order p set to 20. The FFT comparisons of the modeled signals to those of the original signals for these same four data samples are shown in Figures 5 through 8. The smoother of the two lines represents the FFT of the modeled signal. Only the frequencies up to half of the sampling rate are shown. The modeled FFT follows very closely to that of the original for all samples with p = 20. Attempting much higher orders, i.e., values of p = 40 and 50, resulted in almost no change from smaller order values, but added an increase DC bias above the original signal. For practical application purposes we decided to use a reasonable value and a model order of 20 was selected.

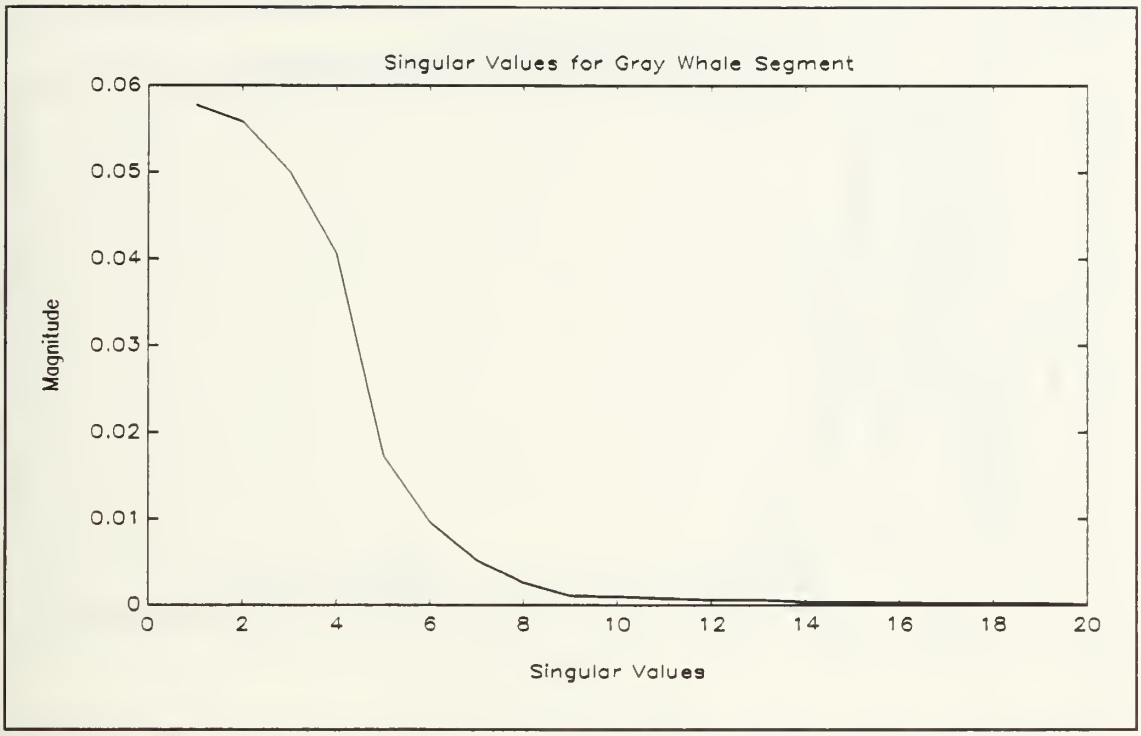

Figure 1: Singular Values of Gray Whale Data Segment.

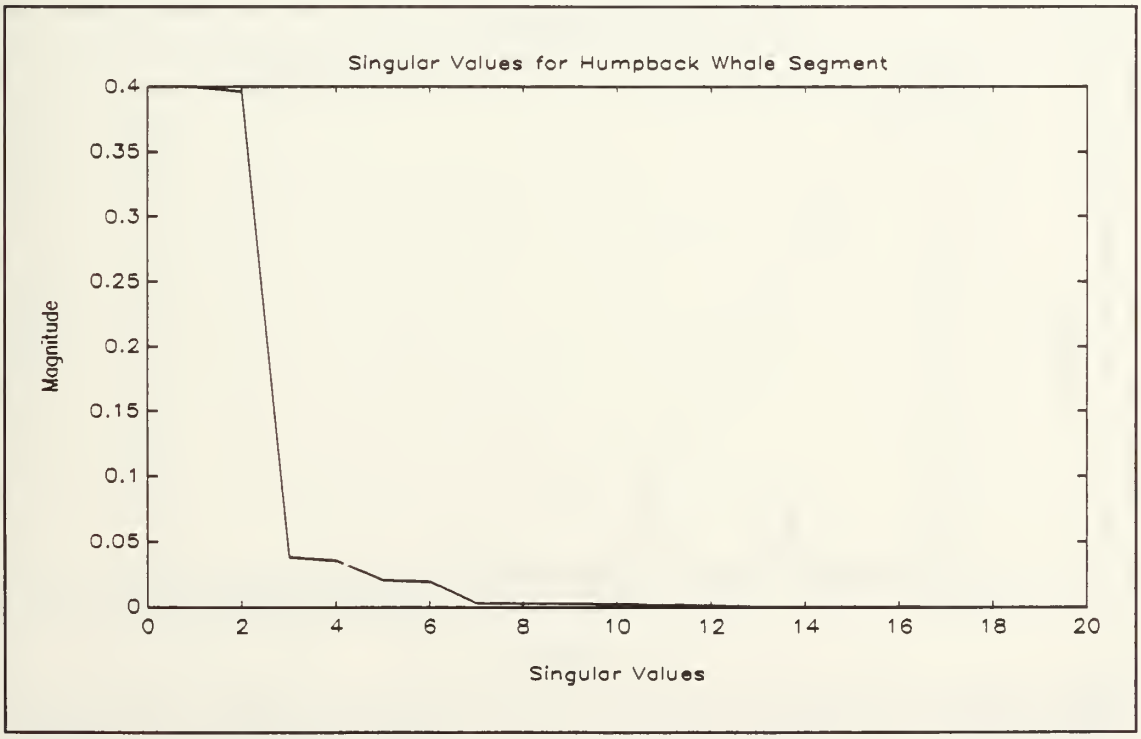

Figure 2: Singular Values of Humpback Whale Data Segment.

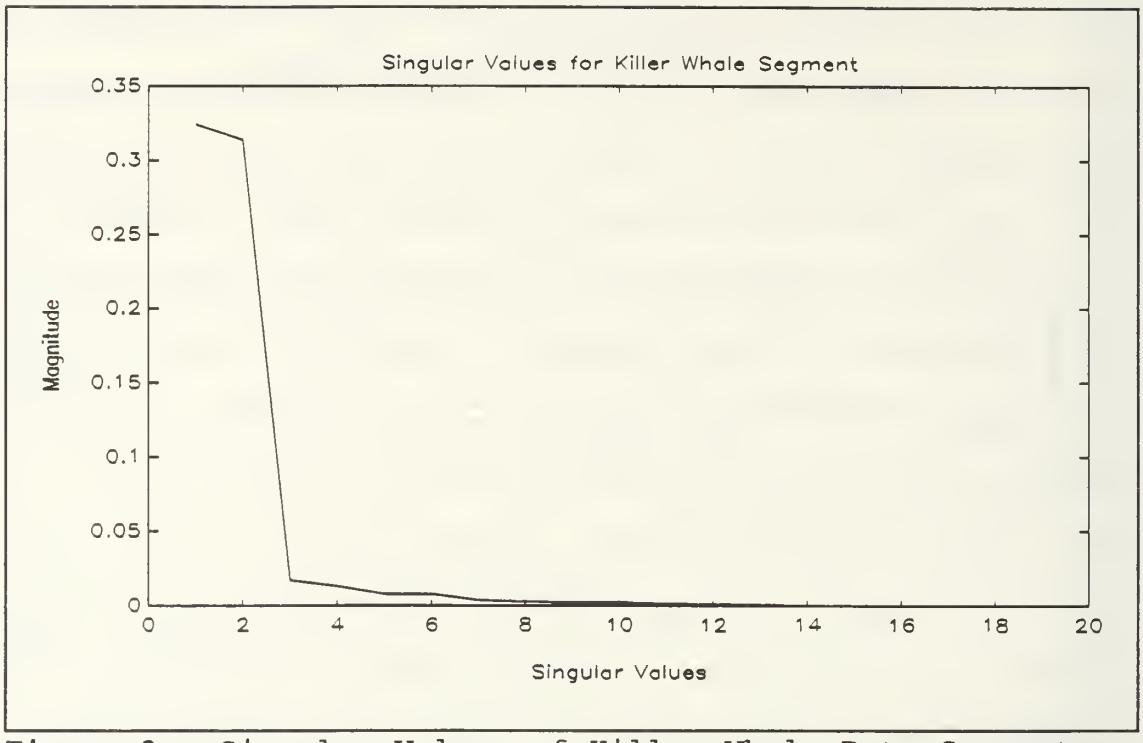

Figure 3: Singular Values of Killer Whale Data Segment.

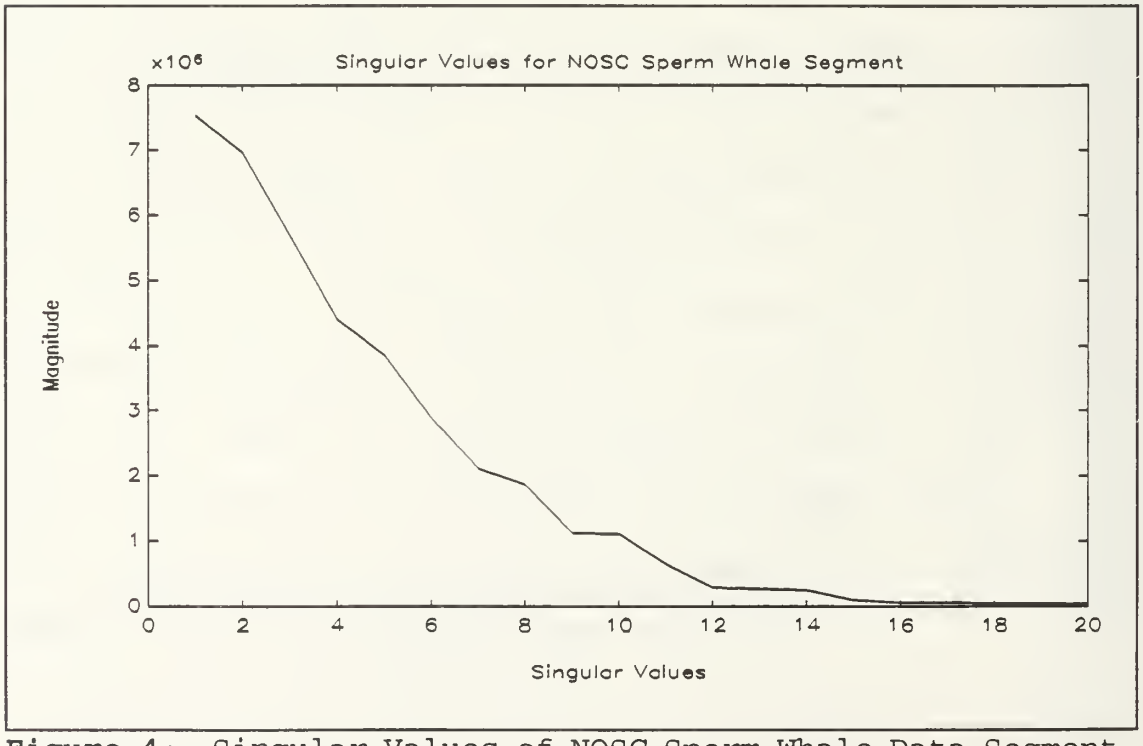

Figure 4: Singular Values of NOSC Sperm Whale Data Segment.

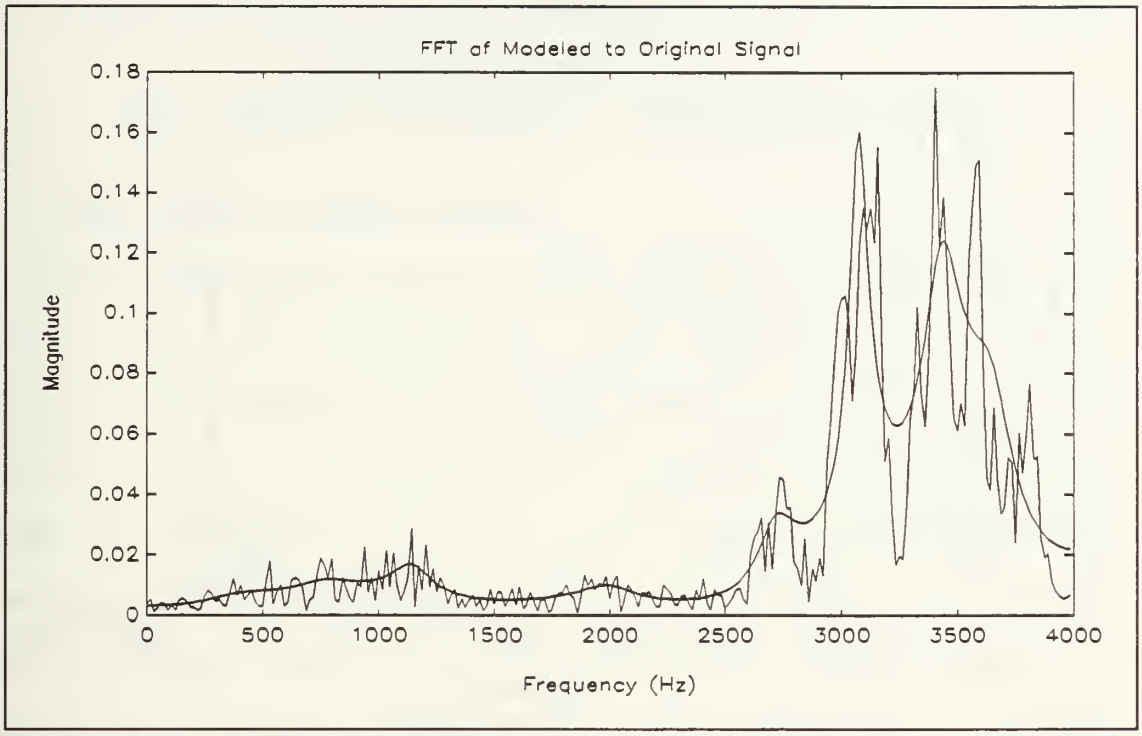

Figure 5: FFT of Modeled to Original Gray Whale Data Segment.

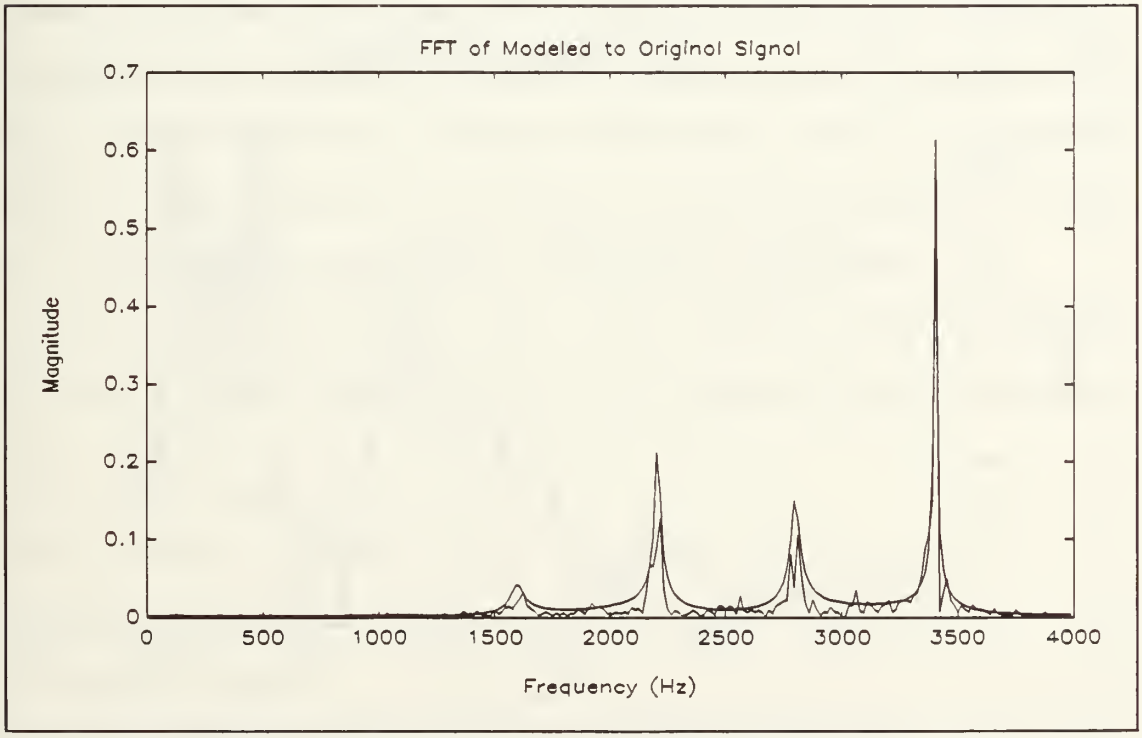

Figure 6: FFT of Modeled to Original Humpback Whale Data Segment.

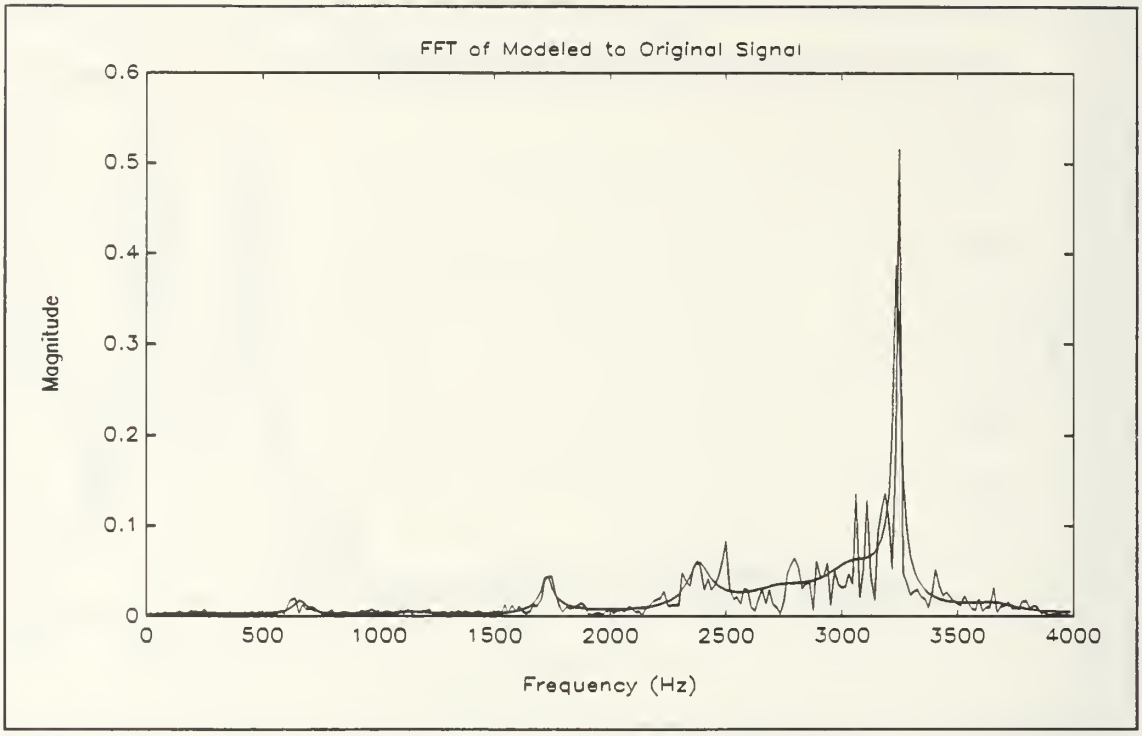

Figure 7: FFT of Modeled to Original Killer Whale Data Segment.

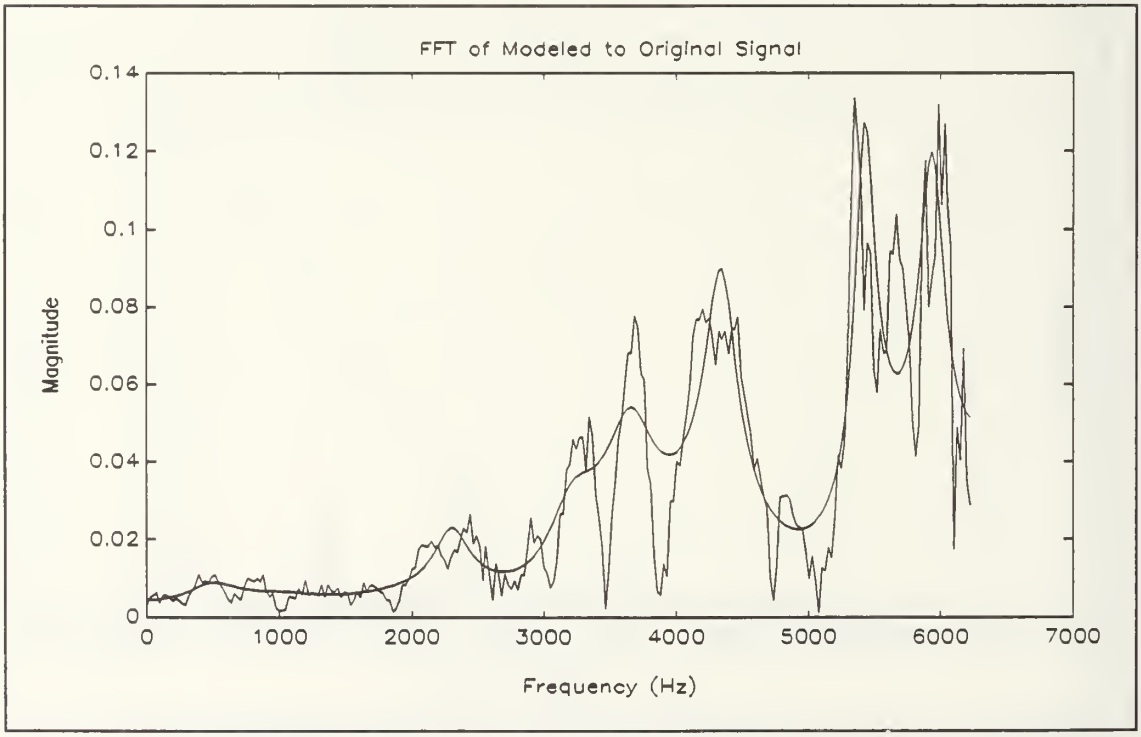

Figure 8: FFT of Modeled to Original NOSC Sperm Whale Data Segment.

III. MODEL DEVELOPMENT OF BIOLOGICAL SIGNALS

A. TIME DOMAIN SIGNAL SOURCE

The biological data for this thesis was made available from three different sources. The Hopkins Marine Station located in Pacific Grove, California, provided the most reliable data with some additional data files from the Naval Ocean Systems Center (NOSC) and the Naval Underwater Systems Command (NUSC), both located in San Diego, California. Unfortunately, there is no information available about the biological data collected from NOSC and NUSC other than the actual biological sound identifications. This creates some confusion with the Hopkins Data as a killer whale recorded in the Monterey Bay may not sound the same as a killer whale from other ocean waters. Since this is not a biological classification paper, a discussion of the differences in whale and other biologic dialects will not be discussed here.

The data available from the Hopkins Marine Station included a sound recording of the following biologic sounds:

- Bearded Seals
- California Sealions
- Gray Whale
- Humpback Whale
- Killer Whale

This audio cassette was processed into a digitized format by students in a speech processing class [Ref 3]. Table I shows the complete listing of all files, their respective sizes, source and biologic identity used in this thesis. Through the use of MATLAB™, the time domain representation of the Gray, Humpback, and Killer whales are provided in figures 9 through 11. Figure 12 presents the time domain representation for the Sealion data. The frequency spectrum of each of these signals, Bearded Seals, another Killer whale and two sperm whale data files are found in figures 13 through 20. The frequency spectrum was calculated using 512 points and shows only frequencies up to half sampling rate.

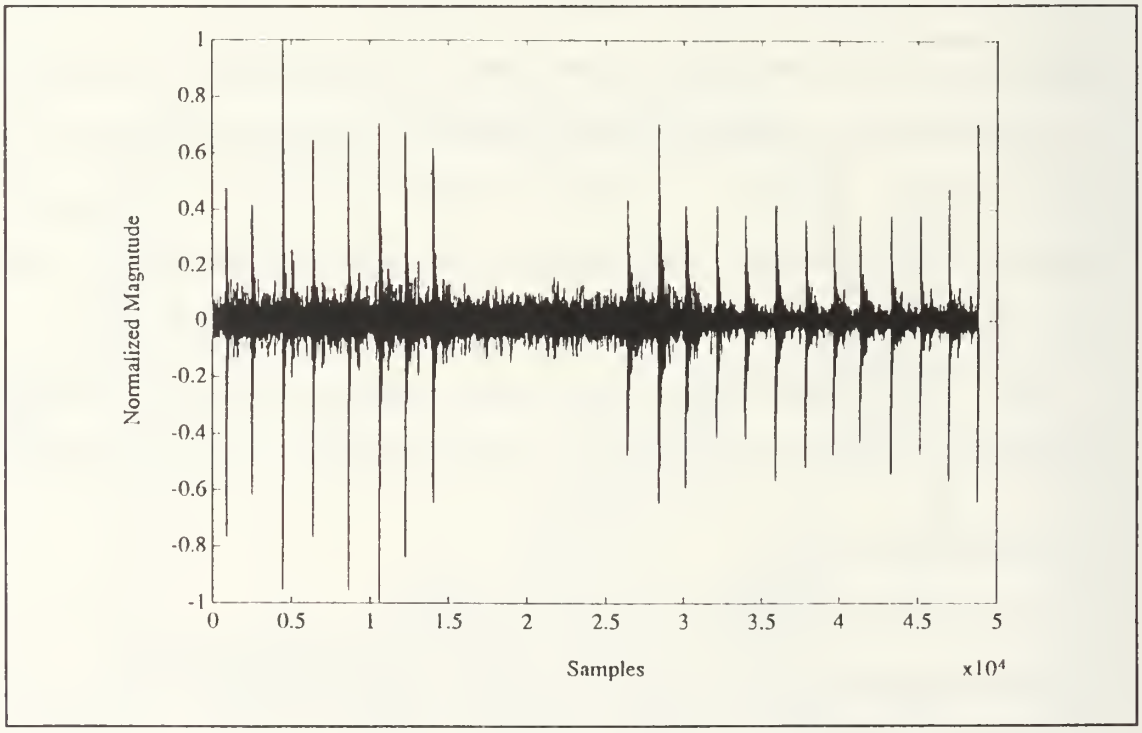

Figure 9: Time Domain Representation of Gray Whale Data.

Table I: TABLE OF BIOLOGICAL DATA.

FILE NAME	SIZE	SOURCE	IDENTITY
BIO_80	300616	NUSC	Ice
BIO_2133a	75000	NUSC	Sperm Whale
BIO_2133b	75000	NUSC	Sperm Whale
BIO_2133c	75000	NUSC	Sperm Whale
BIO_2194a	75000	NUSC	Sperm Whale
BIO_2194b	75000	NUSC	Sperm Whale
BIO_2194c	78616	NUSC	Sperm Whale
BIO_2385	303616	NUSC	Porpoise
BERDED	90148	HOPKINS	Bearded Seals
BERDED2	61859	HOPKINS	Bearded Seals
GRAY	48857	HOPKINS	Gray Whale
HUMPBACK	70063	HOPKINS	Humpback Whale
KILLER	59875	HOPKINS	Killer Whale
ORCA	50001	NOSC	Killer Whale
SEALIONS	33243	HOPKINS	CA Sealions
SPERM	50001	NOSC	Sperm Whale

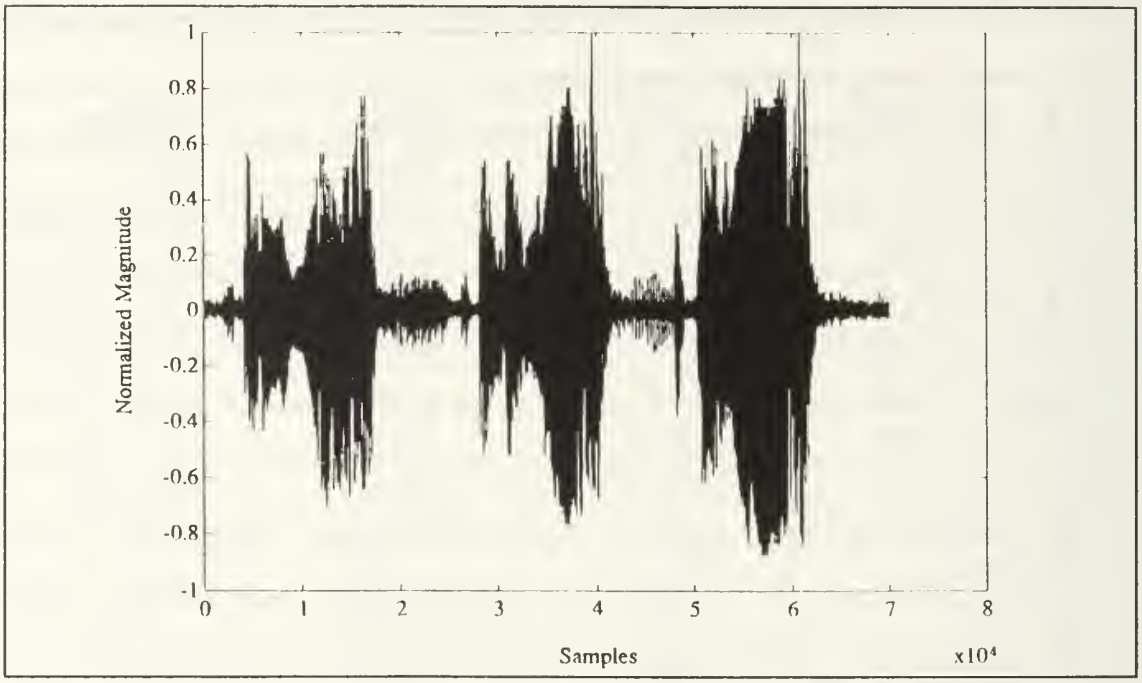

Figure 10: Time Domain Representation of Humpback Whale Data.

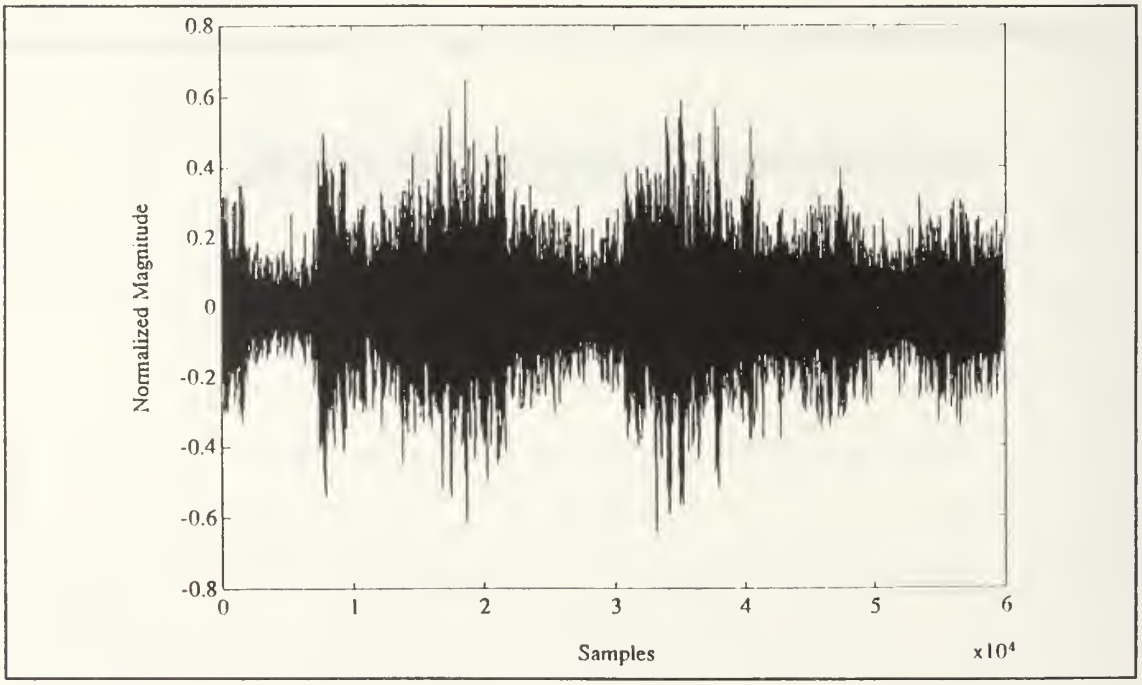

Figure 11: Time Domain Representation of Killer Whale Data.

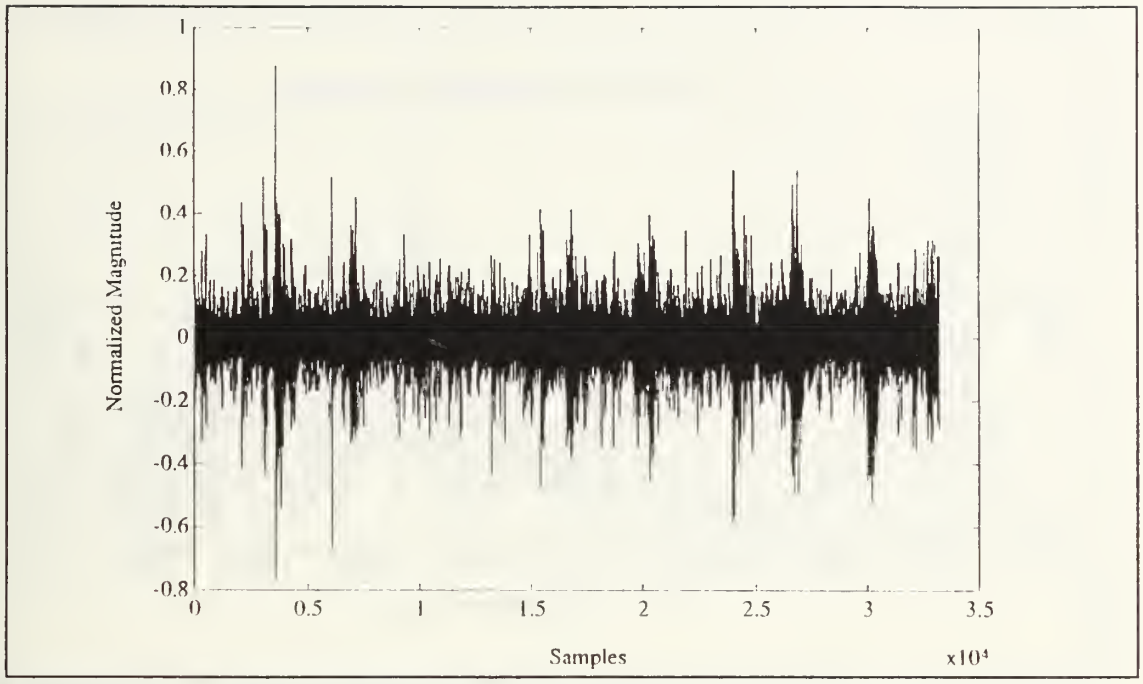

Figure 12: Time Domain Representation of Sealion Data.

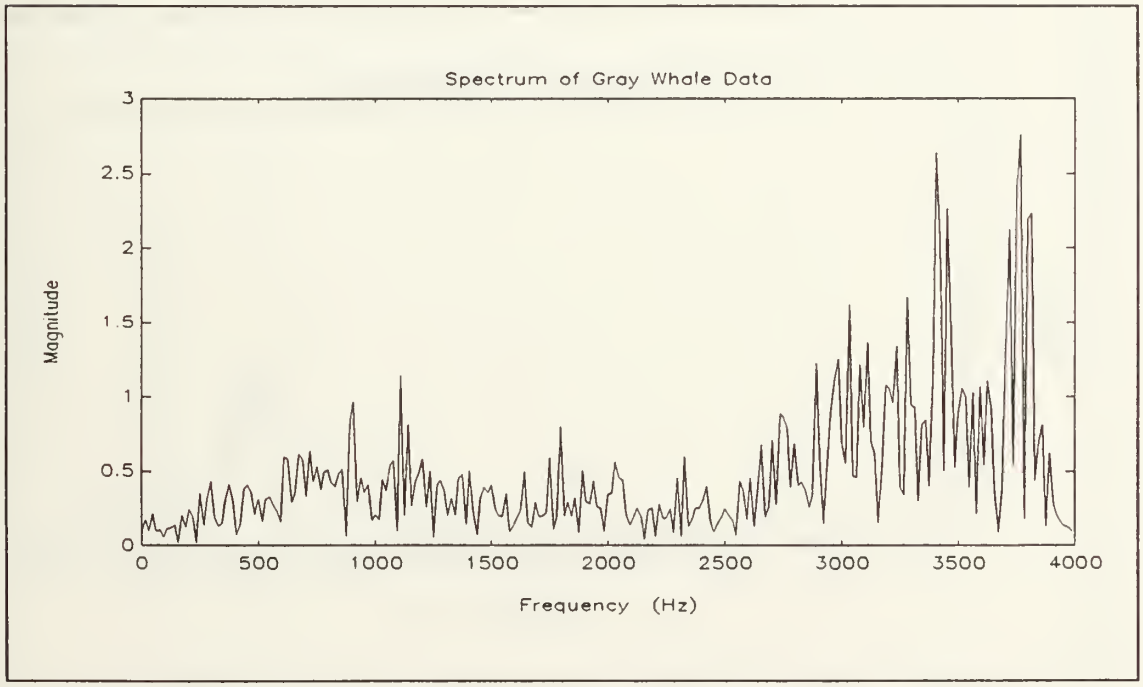

Figure 13: Frequency Spectrum of Gray Whale Data.

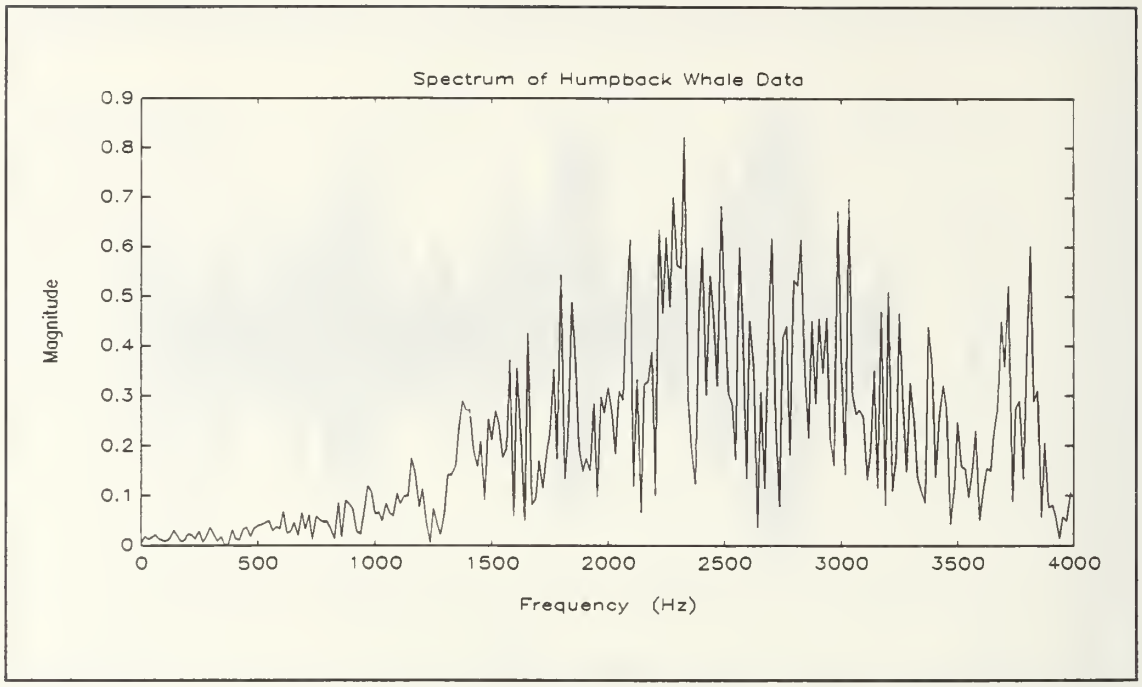

Figure 14: Frequency Spectrum of Humpback Whale Data.

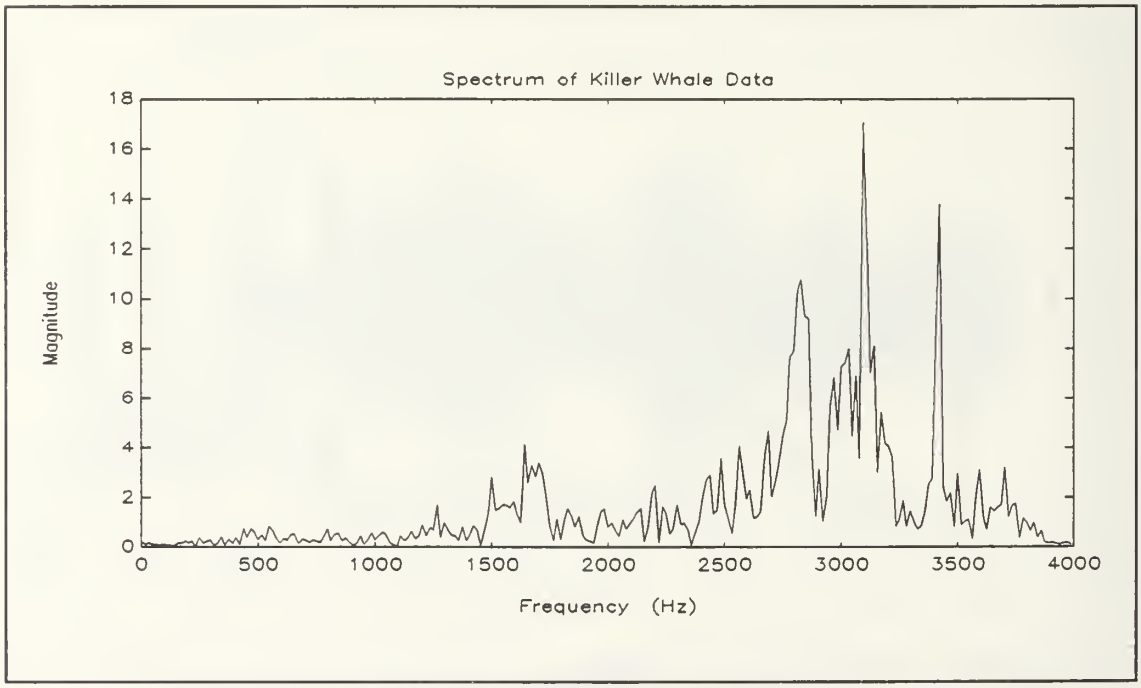

Figure 15: Frequency Spectrum of Killer Whale Data.

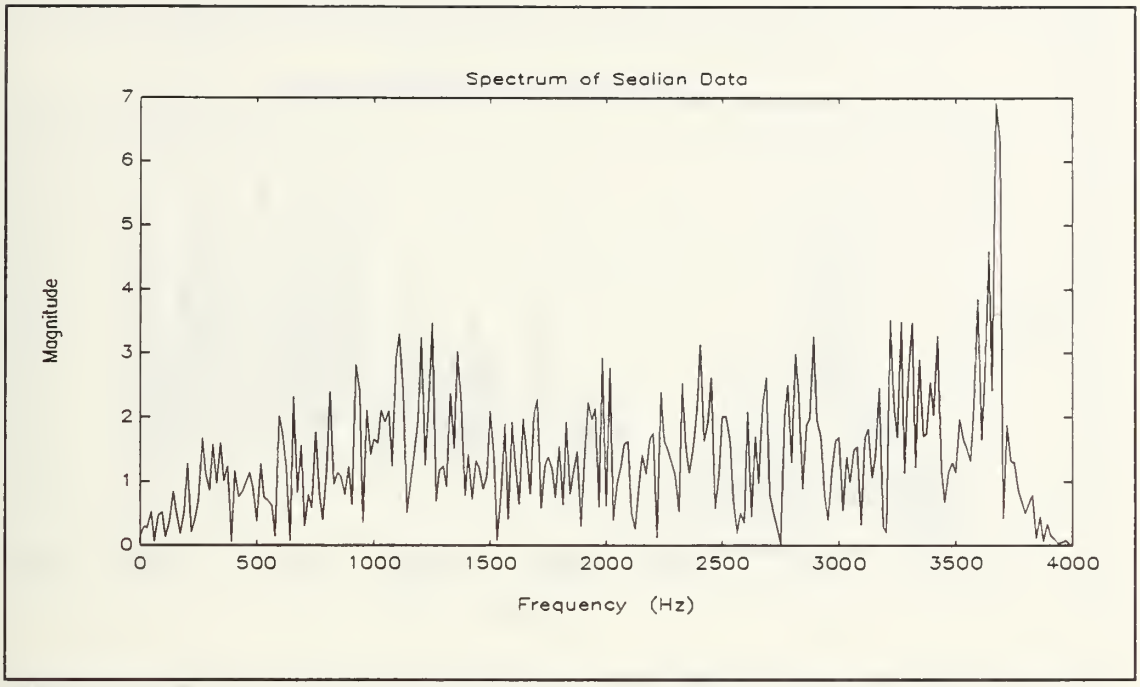

Figure 16: Frequency Spectrum of Sealion Data.

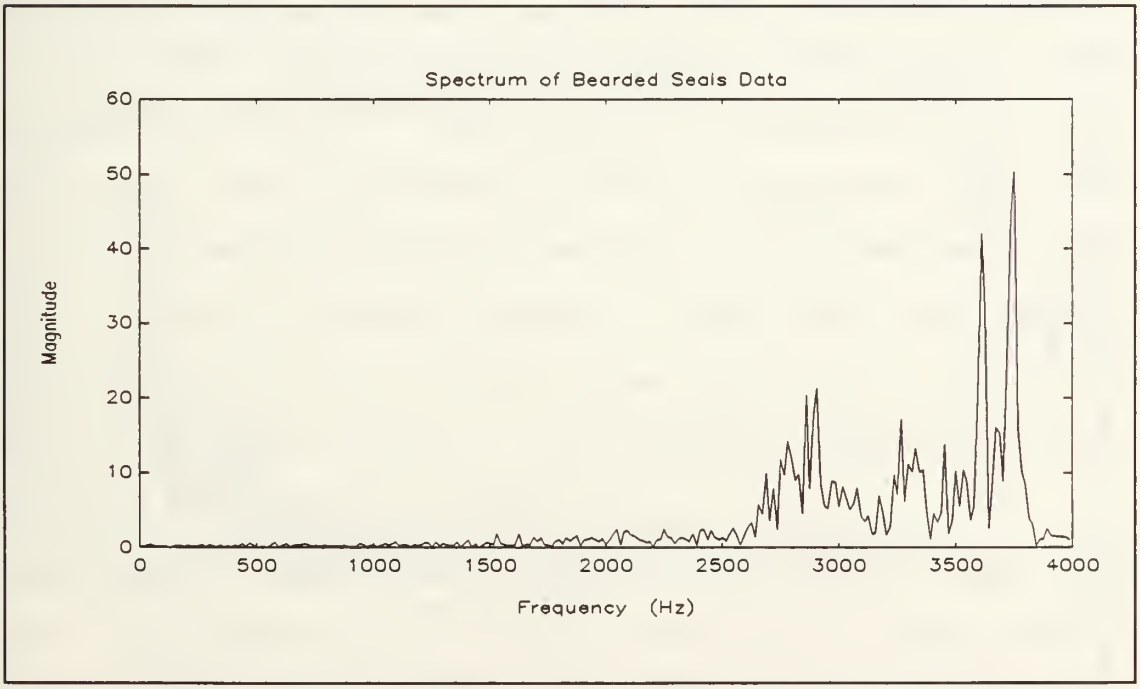

Figure 17: Frequency Spectrum of Bearded Seals Data.

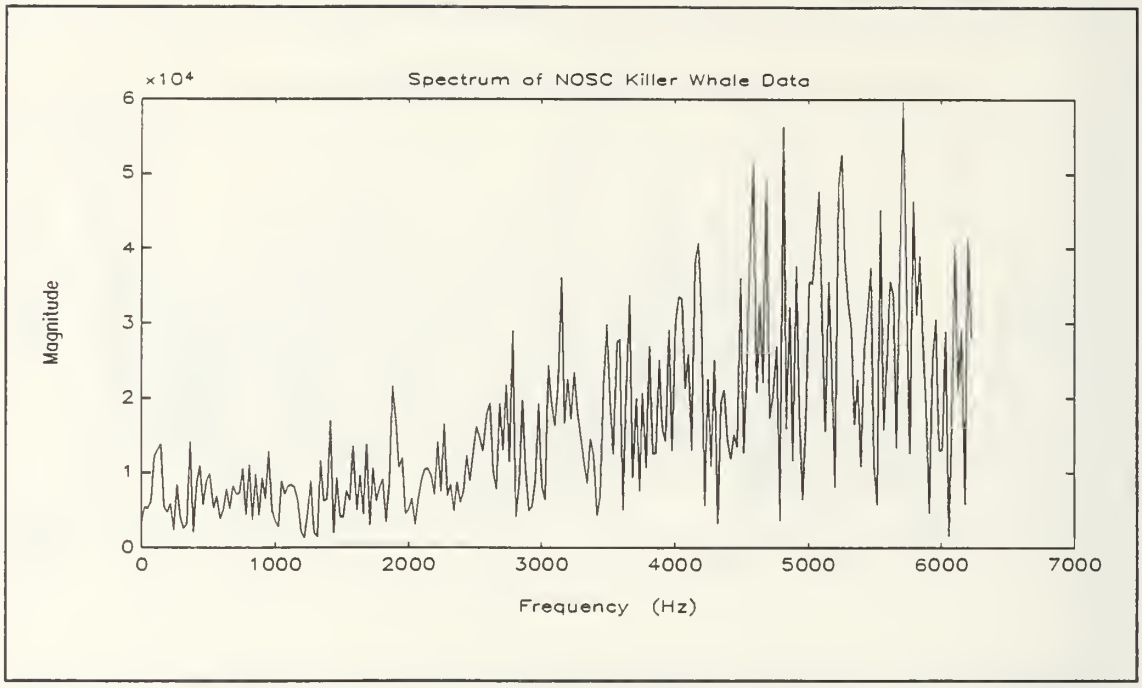

Figure 18: Frequency Spectrum of NOSC Killer Whale Data.

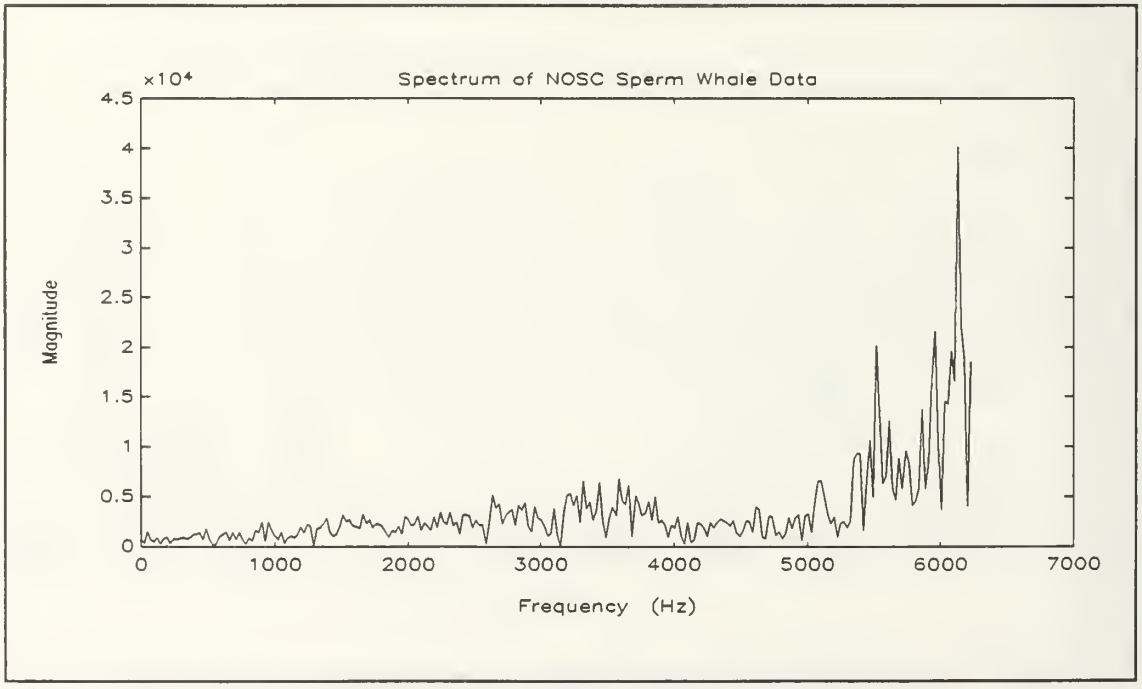

Figure 19: Frequency Spectrum of NOSC Sperm Whale Data.

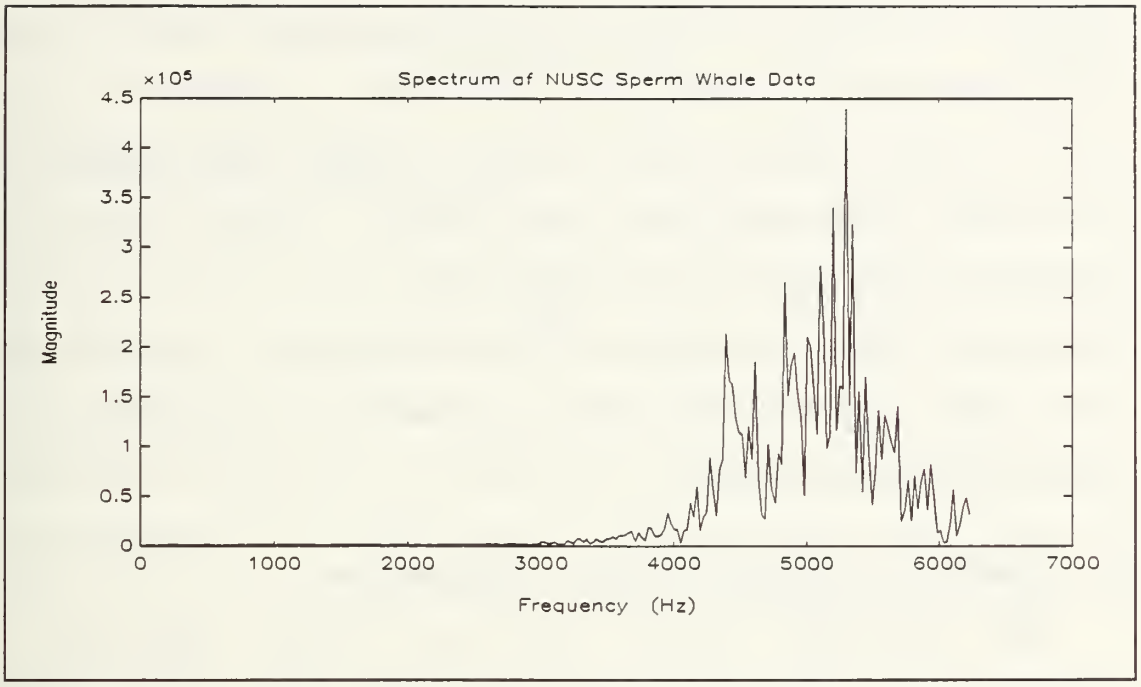

Figure 20: Frequency Spectrum of NUSC Sperm Whale Data.

The biologic data available from NOSC are a sperm whale and killer whale data files. These data files were received in a digitized format and their frequency spectrum are shown in Figures 18 and 19. The spectra of the two killer whales data files, shown in Figures 15 and 18, indicates that there can be a marked difference between the two signatures of this like species of killer whales. Note that the depths, geographical location, sex and even age of the whale can be determining factors as to the different sounds that may originate from the whale.

The same problem occurs with the biological data from NUSC. The majority of this data represents a sperm whale. Since all six of the data files for the sperm whales are

nearly identical, only one frequency spectrum will be shown for comparison with the NOSC sperm data. Figures 19 and 20 are the NUSC sperm whale plots. These two figures are noticeably different from one another. Again without the necessary information about the data, it is difficult to reason why these signatures are so different. Regardless of this fact all this data was utilized for the modeling and classification process.

B. PRONY-SVD METHOD DETAILED DESCRIPTION

The covariance linear prediction normal equations, as described in Chapter II, are solved as part of the least squares Prony method. When noise is present in the data samples, the least squares Prony method estimates of frequency and damping components are usually inaccurate and biased due to the noise effect. Distinguishing weak roots due to noise from the Prony characteristic polynomial is often not realizable. By using both forward and backward linear prediction polynomials, high prediction orders, and SVD the actual exponential signals in the data samples can be identified and accuracy of the estimated frequency and damping components is improved. When the original signal $x(n)$ is noise-free, the following forward linear prediction equation, where $a[0] = 1$, can be used to model p exponentials

$$\sum_{m=0}^{p} a[m]x[n-m] = 0. \tag{3.1}$$

The characteristic polynomial

$$A(z) = \sum_{m=0}^{p} a[m]z^{-m} \tag{3.2}$$

with roots at $z_k = \exp(s_k)$ for $k = 1$ to $k = p$, where $s_k = (\alpha_k + j2\pi f_k)T$ contains the damping factor and frequency of the kth exponential. The same p exponentials may also be generated in reverse time using the backward linear prediction

$$\sum_{m=0}^{p} b[m]x[n-p+m] = 0 \tag{3.3}$$

in which $b[0] = 1$. The characteristic polynomial for this backward linear prediction equation is

$$B(z) = \sum_{m=0}^{p} b^*[m]z^{m-p}. \tag{3.4}$$

The characteristic polynomial $B(z)$ has roots at $z_k = \exp(-s^*_k) = \exp([-\alpha_k + j2\pi f_k]T)$ for $k = 1$ to $k = p$. When $\alpha_k < 0$, the roots of the forward linear prediction characteristic polynomial $A(z)$ lie inside the unit circle in the z-plane ensuring stability. Deterministic exponentials have these root location characteristics associated with $A(z)$ and $B(z)$ and apply themselves well to AR modeling. [Ref. 4:pp. 1309-1310]

By selecting a linear prediction order much higher than the number of exponentials present in the data, the noise bias can be reduced. The drawback in AR modeling procedures is that they introduce extraneous zeros making it difficult to distinguish between zeros of the exponential signals and those due to the noise. Therefore by examining the polynomial roots of both $A(z)$ and $B(z)$, instead of just the roots from either the forward or backward equations separately, the true signal zeros can be distinguished better from the noise zeros. Furthermore, an improvement in the noise reduction is achieved by applying a SVD technique to the covariance matrix of the linear prediction normal equations. Singular values resulting from the noise are constant and very small. The noise singular values are set to zero and only the portion of the singular value matrix S containing signal singular values is inverted in the computation of the pseudoinverse, as described in Chapter II. Had this technique not been applied, the noise singular values would result in extremely large values in the inverse of the matrix S. This would result in erroneous estimates of the model parameters. [Ref. 4:p. 1310]

C. SOFTWARE IMPLEMENTATION OF THE PRONY-SVD METHOD

The main objective of this thesis is to model the biologic data using a Prony-SVD based AR modeling technique. The MATLAB ™code used to implement the Prony-SVD method is found

in Appendix B and a detailed explanation of the software implementation is developed within this section.

The parameters of the AR model $\underline{\hat{a}}$, a vector of model length p, need to be determined from the biologic data vectors. This can be done through two methods; batch processing or recursive processing. The batch processing method takes data in its entirety or in segments. The recursive process yields estimates at every data point, in line with a Recursive Least Squares (RLS) technique. The batch process is used for the Prony-SVD method in this thesis.

Through the use of the batch process a set of parameters $\underline{\hat{a}}$ are determined by a least squares technique. On this premise, a predictor for the observed signal $x(t)$ is generated based on past data and the model parameters $\underline{a}$. The vector parameter $\underline{a} = [a_1 \ldots a_p]$ which minimize the prediction error $\mathbf{E}\left[\left| x(t) - x(t)\right|^2\right]$, where $x(t)$ is the estimated signal obtained from the vector parameter $\underline{a}$. The minimization corresponds to solving a linear set of equations with more equations than unknowns. For all t we have the following equations

$$\hat{x}(t) = -a_1 x(t-1) - a_2 x(t-2) - \ldots - a_p x(t-p) \qquad (3.5)$$

for all $p + 1 < t < N$, where N is the length of the data and p is the order of the model. Furthermore, we assume that $N >> p$. Therefore, the following linear equations are solved for $\underline{a}$

$$-\begin{Vmatrix} x(p) & x(p-1) & \cdots & x(1) \\ x(p+1) & x(p) & \cdots & x(2) \\ \vdots & \vdots & \ddots & \vdots \\ x(N) & x(N-1) & \vdots & x(N-p) \end{Vmatrix} \begin{Vmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{Vmatrix} = \begin{Vmatrix} x(p+1) \\ x(p+2) \\ \vdots \\ x(N) \end{Vmatrix}. \qquad (3.6)$$

The above matrix equation can be rewritten in matrix form as

$$X\underline{a} = \underline{x}, \qquad (3.7)$$

where X has more rows than columns, and the equality is satisfied in the least squares sense. The SVD solution can be derived by writing Equation (3.7) as:

$$X^T X\underline{a} = X^T\underline{x} \qquad (3.8)$$

and decomposing X^TX as

$$X^T X = USV^T, \qquad (3.9)$$

where U and V are orthonormal matrices and S = diagonal$(\sigma_1^2, \ldots \sigma_p^2)$. Usually the singular values σ_i^2 are ordered as $\sigma_1^2 \geq \sigma_2^2 \geq \ldots \geq \sigma_p^2 \geq 0$. The smallest singular values are due to noise and are set to zero. Next, the pseudoinverse of X^TX is used to solve for the model parameters. The SVD-Least squares solution of Equation (3.7) is $\underline{a} = (X^TX)^{-1} X^T\underline{x}$. The vector $\underline{a}$ represents the parameters used to model a segment of the biological data. The vector $\underline{a}$ is the input to the neural network and is used to develop the Itakura distance for classification.

IV. CLASSIFICATION METHODS

A. NEURAL NETWORKS

Through modeling the biologic data signals are reduced from large data segments into a much smaller number of parameters. In this section we address the problem of using these parameters in order to provide an effective classification of the original signals. In particular, a classifier based on neural networks will be developed and tested with the available data. A formal definition of a neural network is given below:

> *A neural network is a parallel, distributed information processing structure consisting of processing elements (which can possess a local memory and can carry out localized information processing operations) interconnected via unidirectional signal channels called connections. Each processing element has a single output connection that branches ("fans out") into as many collateral connections as desired; each carries the same signal – the processing element output. The processing element output signal can be of any mathematical type desired. The information processing that goes on within each processing element can be defined arbitrarily with the restriction that it must be completely local; that is, it must depend only on the current values of the input signals arriving at the processing element via impinging connections and on values stored in the processing element's local memory. [Ref. 5:pp. 2-3]*

1. Neural Network Description and Applicability

From the definition above the basic unit of a neural network is the processing element. A widely accepted configuration is shown in Figure 21. Each processing element has several input connections linearly combined by weighting factors. The results from the linear combination is the

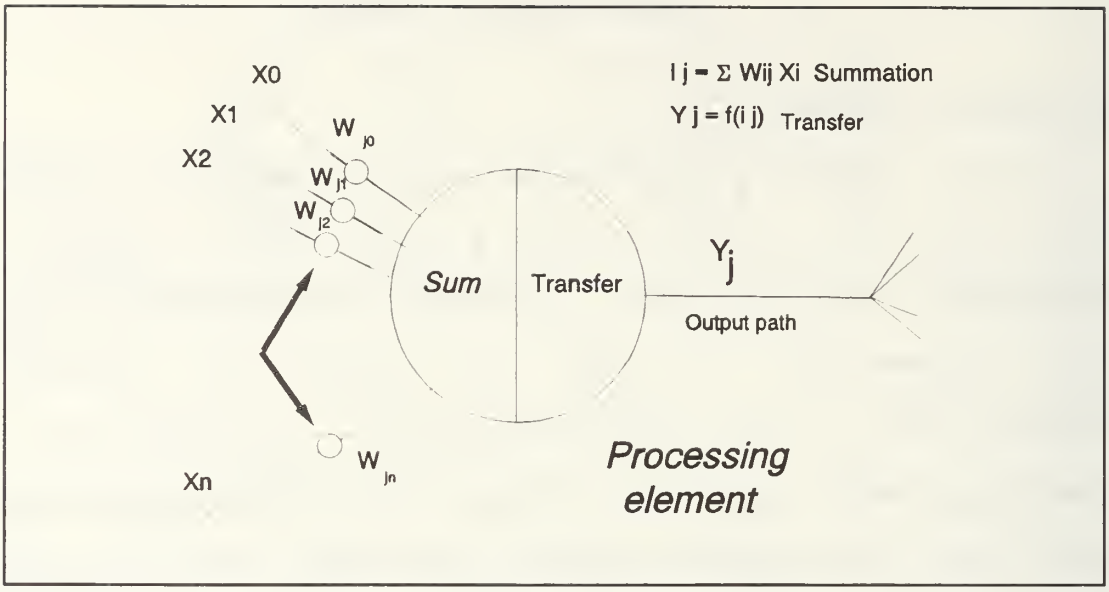

Figure 21: Generic Processing Element

transformed through a nonlinearity (called the transfer function) to form the output of the computing element. Each output of the computing elements is fanned out to several output paths which become inputs to other processing elements. Formally, if we call Y the output and $x_1, x_2, \ldots x_n$ the inputs of a processing cell, we can write

$$Y = f\left(\sum_i w_{ij} x_i\right), \qquad \textbf{(4.1)}$$

where w_{ij}'s are defined as the connection weights.

The neural network consists of many processing elements joined together, grouped into layers. A typical network consists of several layers with full or random connections between successive layers. Typically, networks have two layers with connections to the outside world; an input buffer where data is entered and an output buffer where

the response of the network to the given network is stored. In between these input and output layers are the hidden layers, distinct from the input and output buffers. A simple neural network architecture is shown in Figure 22.

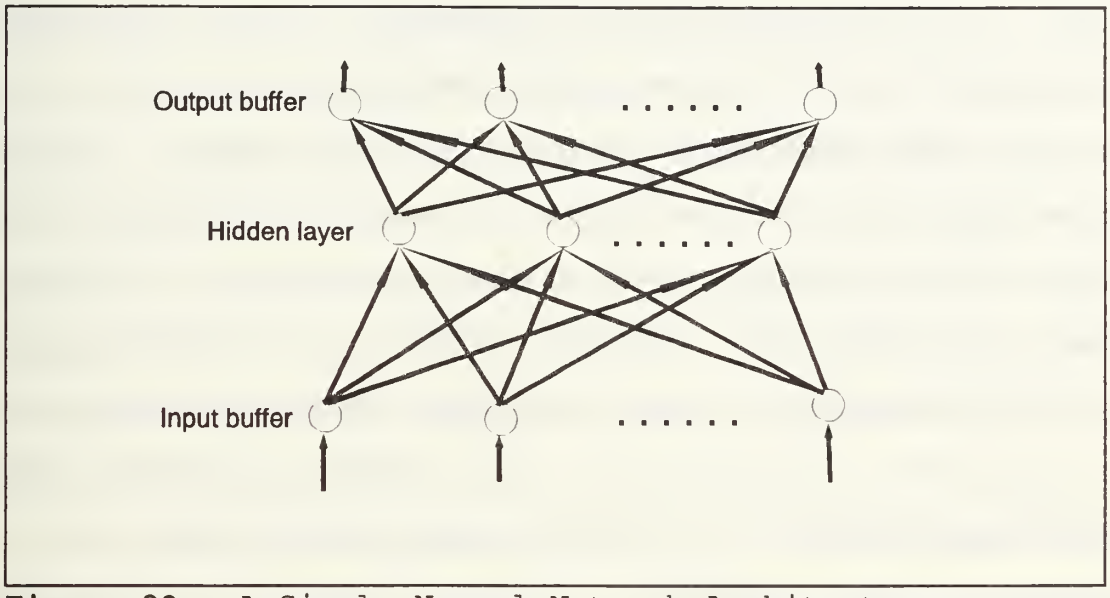

Figure 22: A Simple Neural Network Architecture.

The elements in the input layer have a different structure. They receive only one input each with the connection weight set to unity, and their transfer function is the identity. Their purpose is only to fan the input data out to each of the processing elements in the first hidden layer.

The feedforward network shown is Figure 22 is the simplest form of network, with no feedback connections from layer to the next, or itself. Since there are no feedback loops, its stability is guaranteed.

The network operates in two main phases – Learning and Testing. Learning can be distinguished into a) supervised, b) self-organized, or c) graded.

While the network is learning, the connection weights adapt in response to the data presented to the input buffer and optionally the output buffer. The data presented to the output buffer directly relate to a desired response from the given input data. This type of learning is referred to as "supervised learning". When the desired output is different from the input, the trained network is called "hetero-associative" network. When the output vector is identical to the input for all transactions the trained network is called "auto-associative". In a self-organized training scheme, a network modifies itself in response to *x* inputs. This is also referred to as "unsupervised".

Graded learning (also known as reinforcement training) is a third kind of learning which falls between supervised and unsupervised learning. Learning is reinforced when an external teacher indicates only a good or bad response to the input data. [Ref. 5:pp. 48-49]

The type of learning used specifies its learning rule, i.e., how weights adapt in response to a learning example. During training it may be necessary to show the network thousands of examples. The parameters governing a learning rule may change over time as the network progresses in its

learning. The overall control of the learning parameters is a result of the learning schedule.

Testing or classification, for the purpose of this application, refers to the network's ability to process input data and to create a response at the output.

The other consideration in network performance is whether it operates in synchronous or asynchronous mode. In synchronous mode either all or select groups of processing elements in the system release their output values or fire simultaneously at each time interval. The asynchronous mode allows the processing elements to fire independently of any other processing element.

There are many types of neural networks design for a multitude of applications. Research and practical applications have shown that the back-propagation network works well in pattern classification [Ref. 6:p. NC-104]. For the purpose of biological classification where we map model parameters into classes of signals, a back-propagation network seems to be the logical choice.

2. Back-Propagation Network

The back-propagation network is categorized as a hetero-associative network. It always has an input layer, an output layer and at least one hidden layer. There is no theoretical limit to the number of hidden layers but typically there are no more than 2 hidden layers. The number of

processing elements in the hidden layer will generally be no more than the total number of processing elements in the input and output layers. Utilizing a fully-connected, feedforward network with one hidden layer, as seen in Figure 23, the guidelines for deciding the number of processing elements in the hidden layer are determined by the complexity and the amount of training data available. A couple of basic rules to follow are:

1. The more complex the relationship between the input data and the desired output, the more processing elements are required.

2. A guideline for the upper bound of the number of processing elements in the hidden layer can be determined from the amount of training data available.

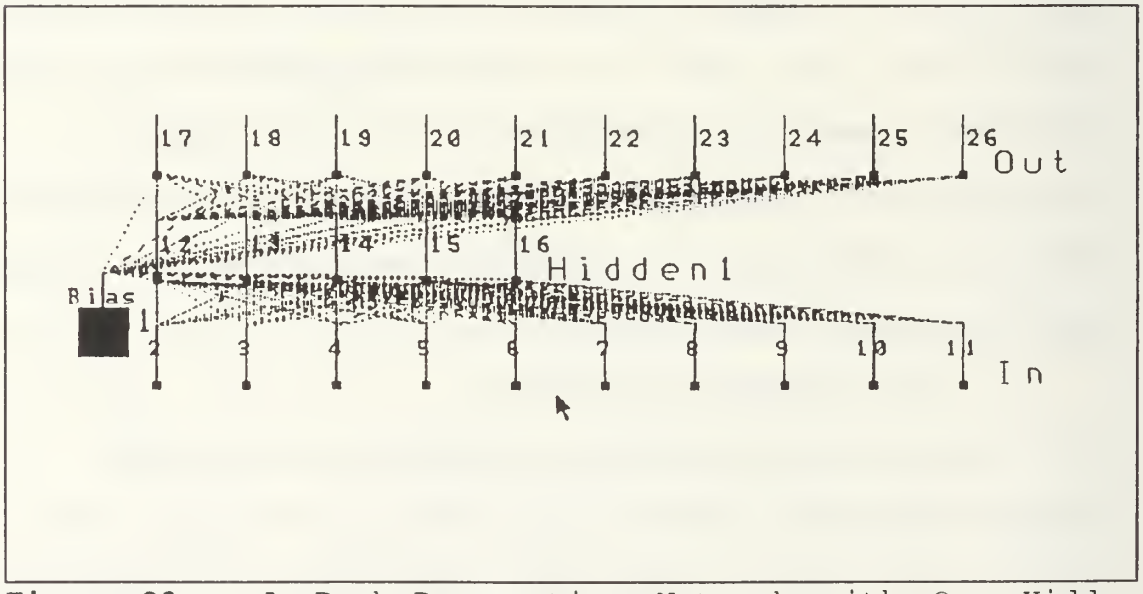

Figure 23: A Back-Propagation Network with One Hidden Layer.

It can be shown that, using a rule of thumb, the number h of hidden layers can be given by the following formula

$$\frac{cases}{10 * (m+n)} = h, \qquad (4.2)$$

where *cases* is the number of training cases,and m and n are the numbers of processing elements in the input and output layers [Ref. 6:p. UN18]. The more random the input data are, the larger the number of processing elements are required in the hidden layer. Note that these guidelines must be tempered by the amount of training data available and the practical size of the network.

The back-propagation network utilized for classification from the NeuralWare Professional II/PLUS software employs a supervised, delta-rule learning scheme. The input data and corresponding output are presented to the system which in turn reduces the error between the actual output of each processing element and the desired output. Gradually the weights are modified via the learning rule to achieve the desired input/output mapping. The delta-rule learning rule and the processing element's transfer function will be described in detail later in this chapter.

In this section, we make use of the following notations:

- $x_j^{[s]}$ ≡ current output of j^{th} neuron in layer s

- $w_{ji}^{[s]}$ ≡ connection weights joining i^{th} neuron in layer [s-1] to j^{th} neuron in layer s
- $I_j^{[s]}$ ≡ weight summation of inputs to j^{th} neuron in layer s

The mathematical process for a single back-propagation element is:

$$x_j^{[s]} = f\left(\sum_i (w_{ji}^{[s]} * x_i^{[s-1]}\right) = f\left(I_j^{[s]}\right), \qquad \textbf{(4.3)}$$

where f is a "squashing" function, namely a sigmoid or the hyperbolic tangent function. Given that the network has a global error function **E**, the error between the actual and desired output is the critical parameter that is fed back through the layers and is defined as:

$$e_j^{[s]} = -\partial E / \partial I_j^{[s]}, \qquad \textbf{(4.4)}$$

where $e_j^{[s]}$ is proportional to the correction of the processing element j in layer s. Furthermore, using the chain rule twice yields:

$$e_j^{[s]} = f'\left(I_j^{[s]}\right) \sum_k \left(e_k^{[s+1]} w_{kj}^{[s+1]}\right). \qquad \textbf{(4.5)}$$

The main mechanism in the back-propagation network is to forward the input to the output, determine the error at the output, then propagate the errors back using the above equations. The goal of the learning process is to minimize the global error by modifying the weights, given the knowledge of the local errors at each processing element. This is

accomplished by using the gradient rule which dictates the weights to change in the direction on the minimum error.

$$\Delta w_{ji}^{[s]} = -lcoef * (\partial E / \partial w_{ji}^{[s]}), \tag{4.6}$$

where *lcoef* is the learning coefficient and $\Delta w_{ji}^{[s]}$ is the connection weight change between the i^{th} neuron in layer s and the j^{th} neuron in layer (s-1). The coefficient $w_{ji}^{[s]}$ is then updated according to the learning rule. The net result is that each weight is changed according to the size and direction of negative gradient on the error surface. Again applying the chain rule we obtain:

$$\begin{gathered} \partial E / \partial w_{ji}^{[s]} = (\partial E / \partial I_j^{[s]})(\partial I_j^{[s]} / \partial w_{ji}^{[s]}) = -e_j^{[s]} x_i^{[s-1]} \\ \Delta w_{ji}^{[s]} = lcoef(e_j^{[s]} x_i^{[s-1]}). \end{gathered} \tag{4.7}$$

References 5 and 6 provide further details of the back-propagation network.

a. Delta Rule Learning Law

The computation of the output error defines the delta rule. The learning law controls the rate of learning in a neural network. When the error is minimized the network has learned. The error is computed as the difference between the desired output and the actual output. The error is transformed by the derivative of the transfer function and propagated back through to the previous layer where it was compiled. This error then becomes the error for that previous layer. The back-propagation of the error continues through

each prior layer until it reaches the input layer. This process is the source for the network's name. The delta rule was chosen as the learning law for the application considered in this work. [Ref. 6:p. RF-195]

The weight update equations for the delta rule are as follows:

$$w'_{ij} = w_{ij} + C_1 * e_i * x_{ij} + C_2 * m_{ij}$$
$$m'_{ij} = w'_{ij} - w_{ij} \qquad (4.8)$$

where w' is the updated weight from the learning rule, m is the present memory and m' is the resulting memory change. The weights are changed in proportion to the error e and the input to that particular connection x. The values for C_1 and C_2 should start between zero and one. Experience has shown the best value to start C_1 at is 0.5 [Ref. 6:p. RF-195]. Should the network increase in size, i.e. number of processing elements, these values should decrease but maintain the same ratio. As the learning process continues, decreases in C_1 and C_2 lead to faster convergence of the network. [Ref. 6:pp. RF-195&196]

b. Transfer Function

Each processing element possesses a transfer function that uses the local memory and the input signal to produce the output signal. Even though the sigmoidal transfer

function is predominantly used in back-propagation networks, the hyperbolic tangent transfer function was chosen for the biological classification network. Figure 24 provides a pictorial view of the hyperbolic tangent transfer function.

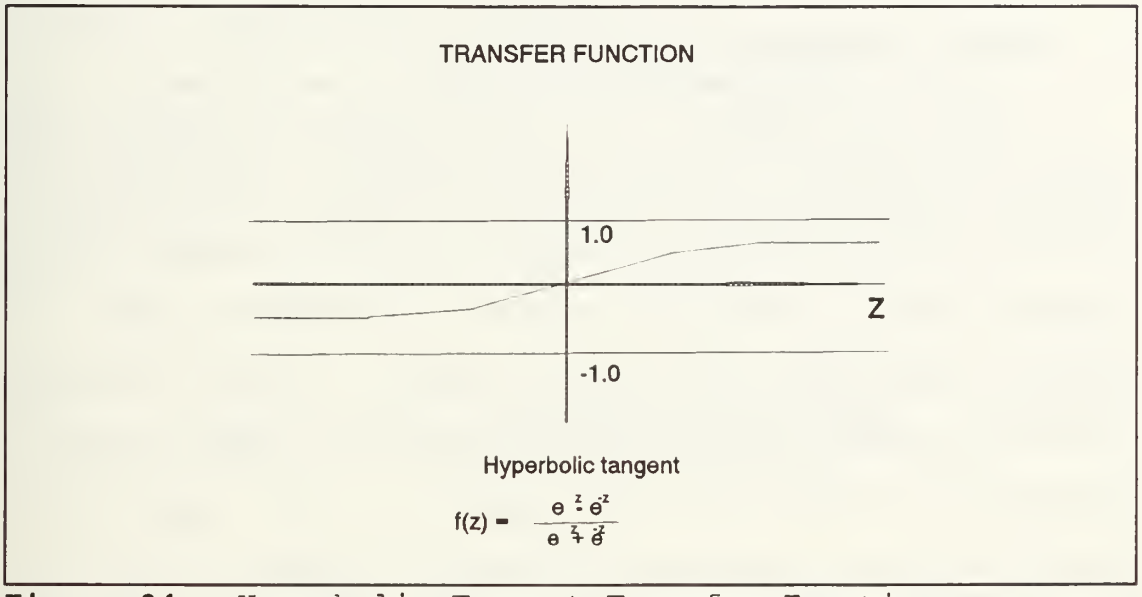

Figure 24: Hyperbolic Tangent Transfer Function.

c. Back-Propagation Network Summary

Back-propagation networks built to classify biological data consist of either three or four layers, to include an input, either one or two hidden layers and an output layer. Various configurations were utilized until approximately seven of all the various combinations were found to have the best convergence criteria. All networks have 20 input processing elements. These 20 processing elements represent the 20 model parameters discussed in Chapter II. The number of output processing elements was set at either two or five. The two outputs correspond to a classification of

"Whale" or "Not a Whale" while the five outputs reflect the determination of one of four species of whale or some other biologic. All the results from these networks are found in the following chapter.

B. ITAKURA DISTANCE CLASSIFICATION

A different approach to classification is investigated with the concepts of distance measurements. The Symmetrized Itakura distance measured between a known reference and an unknown signal is used as to classify the unknown signal. The distance measurement is used to compute a scalar difference between waveforms, and to measure appropriate similarities (or differences) between the two spectra. This leads to the definition of an appropriate "distance" measure which can be used for classification.

In particular, given any two spectra $f(\omega)$, $g(\omega)$ we look at a measure of their similarities by a scalar $d(f,g)$ with the following properties:

1) $d(f,g) \geq 0$ for all f,g
2) $d(f,g) = 0$ if and only if $f(\omega) = g(\omega)$ almost everywhere
3) $d(f,g) = d(g,f)$
4) $d(\alpha f,\beta g) = d(f,g)$ for any nonzero scaling factor $\alpha,\beta \epsilon \mathbb{R}$.

Furthermore, d should be computationally feasible in real time, and have some intuitive appeal also.

An important class of distance measurements is based on spectral ratios, previously used to classify speech signals [Ref. 7]. In the next section we develop a particular distance measure based on spectral ratios which is the basis of our next classification scheme.

1. Symmetrized Itakura Distance Measure

Given any two spectra $f(\omega)$, $g(\omega)$ such that $f(\omega)$, $g(\omega) \neq 0$ where $\omega \epsilon \mathbb{R}$, we define the Symmetrized Itakura (SI) distance as:

$$d_{SI}(f,g) = \ln \frac{1}{2\pi} \sqrt{\left(\int_{-\pi}^{\pi} \frac{f(\omega)}{g(\omega)} d\omega\right) \cdot \left(\int_{-\pi}^{\pi} \frac{g(\omega)}{f(\omega)} d\omega\right)}. \qquad (4.9)$$

We can show that Equation (4.9) exhibits the properties of a distance by the following:

1. $d(f,g) = d(g,f)$, this is obvious from the above expression.

2. $d(f,g) \geq 0$, and $d(f,g) = 0$ if and only if $f(\omega) = kg(\omega)$ for all ω, and a scalar $k \neq 0$.

Proof: Recall that the Schwartz Inequality is given by:

$$\left|\int x(t)\, y^*(t)\, dt\right|^2 \leq \left(\int |x(t)|^2 dt\right)\left(\int |y(t)|^2 dt\right), \qquad (4.10)$$

for any function $x(t)$, $y(t)$ or more appropriately $x(\omega)$, $y(\omega)$. Replacing $x(t)$ and $y(t)$ with $f(\omega)$ and $g(\omega)$ in Equation (4.10) leads to:

$$\left(\int_{-\pi}^{\pi}\frac{f(\omega)}{g(\omega)}\,d\omega\right)\left(\int_{-\pi}^{\pi}\frac{g(\omega)}{f(\omega)}\,d\omega\right)\geq\left|\int_{-\pi}^{\pi}\sqrt{\frac{f(\omega)}{g(\omega)}}\sqrt{\frac{g(\omega)}{f(\omega)}}\,d\omega\right|=2\pi. \quad \textbf{(4.11)}$$

The equality of Equation (4.11) holds if and only if

$$\sqrt{\frac{f(\omega)}{g(\omega)}}=k\sqrt{\frac{g(\omega)}{f(\omega)}} \quad i.e. \ |f(\omega)|=k|g(\omega)|. \quad \textbf{(4.12)}$$

Thus, using Equation (4.12) in Equation (4.11) leads to:

$$\ln\frac{1}{2\pi}\left(\int_{-\pi}^{\pi}\frac{f(\omega)}{g(\omega)}\,d\omega\right)\left(\int_{-\pi}^{\pi}\frac{g(\omega)}{f(\omega)}\,d\omega\right)\geq 0, \quad \textbf{(4.13)}$$

where Equation (4.13) equals zero if and only if $f(\omega)=kg(\omega)$. In summary we can say that the Symmetrized Itakura distance is symmetric by its definition and that the distance is insensitive to amplitude scaling. These two necessary properties make this distance measurement desirable for classification. [Ref. 7:pp. 38-39]

a. SI Distance Measures Applied to AR Processes

Autoregressive spectra can be characterized directly in terms of the AR model parameters, referred to as the predictor coefficients. The pth order AR spectrum of the biologic data is defined as

$$S(\omega)=\frac{\sigma^2}{A(e^{j\omega})A(e^{-j\omega})}=\frac{\sigma^2}{|A(e^{j\omega})|^2}, \quad \textbf{(4.14)}$$

where $A(z) = 1+A_1z^{-1}+\ldots+A_pz^{-p}$ represents the polynomial associated with the AR model. The spectrum $S(\omega_k)$ of $A(z)$ is

obtained by computing an FFT of the the coefficient sequence $\{1, a_1, a_2, \ldots, a_p, 0, 0 \ldots\}$ using Equation (4.14), where

$$\omega_k = \frac{2\pi k}{N} \qquad for\ k = 0, 1, \ldots, N-1. \tag{4.15}$$

To accomplish the classification, we choose a model with spectrum $S_i(\omega)$ for i = 1,...,N. Then for any other segment $S(\omega)$ we measure the distance

$$d_{SI}(S_i, S) = \ln\left(\frac{1}{2\pi}\int \frac{S_i(\omega)}{S(\omega)}\, d\omega\right)\left(\frac{1}{2\pi}\int \frac{S(\omega)}{S_i(\omega)}\, d\omega\right) \qquad for\ i = 1, \ldots, N, \tag{4.16}$$

and we classify the segment in the class that has the minimum distance to the reference. Figure 25 provides a block diagram of this classification process.

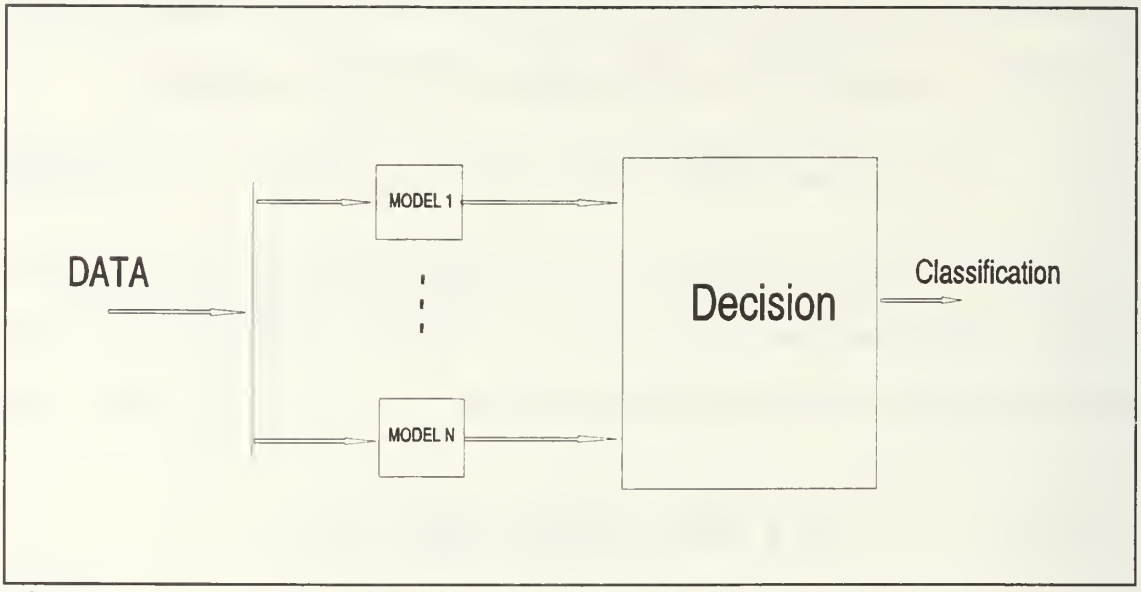

Figure 25: Block Diagram of Symmetrized Itakura Distance Classification Process.

To make the spectrum smoother (less noisy) we use the AR model:

$$S(\omega) = \frac{\sigma^2}{|A(e^{j\omega})|^2}, \qquad \textbf{(4.17)}$$

with $A(e^{j\omega})$ estimated from the Prony-SVD method.

Since we want to do the modeling process digitally, we have two alternatives: a) approximate the Itakura distance using FFT's; b) use the residue theorem. To compute the spectra of $S(\omega)$ we need to consider the AR parameters in terms of a pth order polynomial $A(z)$ where $A(z)=1+a_1z^{-1}+a_2z^{-2}+\ldots+a_pz^{-p}$. We substitute $z = e^{j\omega}$ into the general pth order polynomial $A(z)$ with the result

$$|A(e^{j\omega})|^2 = A(z)A(z^{-1})\,|_{z=e^{j\omega}}. \qquad \textbf{(4.18)}$$

Equation (4.18) can be expanded into

$$A(z)A(z^{-1}) = \sum_{i=0}^{p} \rho_i (z^i + z^{-i}), \tag{4.19}$$

where ρ_i's are called the spectral parameters and are defined by

$$\rho_i = \sum_{k=i}^{p} a_k a_{k-i}. \tag{4.20}$$

Using the residue theorem we know that the integral of a complex function over a closed contour is proportional to the sum of the residues of the function inside the contour. Let r(z) be the ratio of $S_i(\omega)$ and $S(\omega)$ and apply Equation (4.17)

$$r(z) = \frac{S_i(\omega)}{S(\omega)} = \frac{\sigma_{S_i}^2}{\sigma_S^2} \cdot \frac{A_{S_i}(z)A_{S_i}(z^{-1})}{A_S(z)A_S(z^{-1})}. \tag{4.21}$$

Taking the inverse Z Transform of Equation (4.21) at lag 0 leads to:

$$\begin{aligned} r_{S_i/S}(0) &= \frac{\sigma_{S_i}^2}{\sigma_S^2} \cdot \frac{1}{2\pi j} \oint \frac{A_S(z)A_S(z^{-1})}{A_{S_i}(z)A_{S_i}(z^{-1})} dz \\ &= \frac{\sigma_{S_i}^2}{\sigma_S^2} \sum \left(residues\ of\ \frac{r(z)}{z} \right), \end{aligned} \tag{4.22}$$

where $r(z)$ is the ratio of $S_i(\omega)$ and $S(\omega)$ as in Equation 51. The closed contour denotes the integral over the unit circle. This implies that the poles to be taken into account in the computations of the residues are inside the unit circle. [Ref. 7:pp. 42-46]

In considering both of these methods, the complexity of the residue theorem encouraged us to use the FFT method to compute the Symmetrized Itakura distance measurement for classification.

2. Implementing the Symmetrized Itakura Distance

The Symmetrized Itakura distance was implemented using MATLAB™ to classify the biological data signals. The MATLAB™ code for this procedure is found in Appendix B and the description of its implementation will follow.

The Symmetrized Itakura distance measure is computed between a reference $A_{Si}(e^{j\theta})$ and an observation $A_S(e^{j\theta})$, where $S_i(\theta)$ and $S(\theta)$ are the spectra:

$$S_i(\theta) = \frac{1}{\left|A_{S_i}(e^{j\theta})\right|^2} \qquad S(\theta) = \frac{1}{\left|A_S(e^{j\theta})\right|^2}. \tag{4.23}$$

Two vectors are used for the software implementation:

$$\underline{S_i} = \left[\left|A_{S_i}(e^{j0})\right|^2, \left|A_{S_i}(e^{j\frac{2\pi}{N}})\right|^2 \text{----------} \right]$$
$$\underline{S} = \left[\left|A_S(e^{j0})\right|^2, \left|A_S(e^{j\frac{2\pi}{N}})\right|^2 \text{----------} \right], \tag{4.24}$$

where N is the number of points desired for the FFT. For this application 512 points were chosen. The vectors S and S_i were created by taking the parameters derived from modeling the biological signal and placed in the following format

$$A(z) = 1 - a_1 z^{-1} - a_2 z^{-2} - \ldots - a_p z^{-p}, \qquad (4.25)$$

where p is the model order.

To compute the Symmetrized Itakura distance, it is necessary to solve for the quantities $r_{Si/S}(0)$ and $r_{S/Si}(0)$ that have the following integral form

$$\frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{S_i(\theta)}{S(\theta)}d\theta. \qquad (4.26)$$

The above integral is approximately equal to

$$\frac{1}{2\pi}\sum_{k=0}^{N-1}\frac{S_i(\frac{2\pi k}{N})}{S(\frac{2\pi k}{N})}\cdot\frac{2\pi}{N}, \qquad (4.27)$$

where the ratio $2\pi/N$ corresponds to the change in θ.

3. Data Manipulation

The first step to utilize the Symmetrized Itakura distance measure was to establish the reference signals. Due to limited data, it was necessary to take the files for Gray, Humpback, Killer, and NOSC Sperm Whales and the Sealions and separate out the first segment of sound. This segment of sound was then modeled using the Prony-SVD method. An order of 30 was chosen to ensure a more sensitive model. The five modeled reference sound segments were then structured for Symmetrized Itakura distance measures using the MATLAB™ code *ITAREF.M*, that is found in Appendix B. Each modeled reference segment is manipulated in the exact manner as described above.

All five reference mat files were saved as *I?ref.mat* in the MATLAB™ environment, where ? represents one of the five letters; G, H, K, L, and S, for Gray, Humpback, Killer, Sealions, and Sperm Whales respectively.

Next the MATLAB™ code *ITAKURA.M* was used to classify the sounds using the Symmetrized Itakura distance scheme. The code can be found in Appendix B. This m-file loads all five reference files and one of sixteen matrices of Prony-SVD modeled biological data. The matrix must be manipulated so that the FFT can be taken column by column. The net result is a Symmetrized Itakura distance measurement for each segment of the Prony-SVD modeled biological data against each of the Itakura reference signals. The measurement results are found in Chapter V.

V. CLASSIFICATION RESULTS

A. TRAINING THE NEURAL NETWORK

Before the classification results can be discussed in regards to the neural network performance, it is first necessary to review the training operation of the neural network.

The training file utilized to train the neural network was built from the biological data described in Chapter III. Using MATLAB™, a matrix file was constructed with each row representing the 20 model parameters from a segment of biological data. The shortage of available data made it necessary to use the data for both training and testing. Two-thirds of the data files were used to construct the training file. This results in a training file consisting of 1272 cases available to teach the network. This matrix was saved in ASCII format and then transferred to the Sun Workstation where the line editor was used to input the supervised training information. Two training files were created differential in the output results. One training file would classify an input as a "whale" or "not a whale" and the other training file classifies an input as one of four whale species or an other category.

In much the same manner, a testing file was constructed to test the networks performance. The remaining one-third of each biological file plus any additional data that was not

used in the training portion was used to build this testing file. Two testing files were created to correspond to their respective training files. During the testing routine, the network was tested with the entire file, making one complete pass through the file. The accuracy of the test is discussed in the next section.

B. PERFORMANCE OF THE NEURAL NETWORK

Seven different neural networks successfully converged to learn the modeled biological parameters. Even though the simplest networks often work the best, it was found through trial and error that a back-propagation network with two hidden layers converged faster, with fewer iterations than did the networks with only one hidden layer.

As mentioned in the previous chapter, two output configurations were used in the training network. One configuration used five output processing elements to classified four different whales and the fifth was used for others not in the four classifications. The other configuration with only two output processing elements classified for either a "whale" or "not a whale". Note that the network performance is judged during the testing phase. The root mean squared (RMS) error is the criteria that was used to determine both convergence criteria and how well the network tested.

Table II lists the seven network configurations that were used to classify biological data either as a Gray Whale, Humpback Whale, Killer Whale, Sperm Whale or other. The number of processing elements in the input layer and the output layer is not available in the table since those numbers remained constant at twenty and five respectively. The table provides the number of processing elements in each of the hidden layers, the number of iterations that each network went through until the convergence criteria was met and the RMS training error at convergence in addition, the RMS testing error obtained for the testing files is also included in Table II. When using the Neuralware Software, the testing function produces a file that lists the output desired and the actual output for each input case to the network. The program *RMSE.m* takes this file and computes the RMS error of the entire testing file. The resulting RMS testing error indicates whether the network has learned or not based on the data presented during training.

The convergence criteria for each network configuration was set at 0.1. This convergence criteria reflects when the RMS training error is less than a specified threshold level as desired by the network designer. As can be seen in Table II, this convergence criteria was successfully met.

Even more encouraging is the RMS testing error. The RMS testing error is calculated from a line by line comparison for the entire test file of the desired output to the actual

output. With the exception of one configuration, each RMS testing error is considerably smaller than the convergence RMS training error. Furthermore, results in Table II show that the performance of the neural network improves as the number of iterations in the training phase increases.

Table II: NETWORK RESULTS WITH 20 INPUT PE'S AND 5 OUTPUT PE'S.

HIDDEN LAYER #1	HIDDEN LAYER #2	ITERATIONS	TRAINING ERROR	TESTING ERROR
20	0	63248	.0929	.0429
20	5	13036	.0907	.0852
20	15	27008	.0942	.0501
15	0	65933	.0832	.0467
15	5	33750	.0916	.0492
15	10	31338	.0931	.0509
15	15	35297	.0747	.0501

The two networks with iterations of 63,248 and 65,933 had the lowest overall RMS testing errors. These were also the two networks with only one hidden layer. This indicates that additional hidden layers reduce the amount of data necessary to train the network to convergence criteria. Finally, as expected, this table also shows that the networks with the fewest number of iterations had the largest RMS testing errors.

The second configuration used only two output processing elements for classification and provided some interesting results. These networks had the same number of input and hidden layer processing elements as did those with five output processing element networks. The results are listed in Table III in the same format as described for Table II discussed earlier. Overall, it took considerably fewer iterations to meet the convergence criteria, an RMS error of 0.1, than for networks with five output processing elements. Comparing the training error obtained in Table III and in Table II, shows that most of the training errors are much smaller than those in Table II, except for two network configurations. An observation was made earlier that the fewer number of iterations required for convergence the smaller the resulting RMS training error is. However, this is not the case for the two exceptions mentioned before. In Case 1 we used 20 PE's in the Hidden layer number 1 and 0 in the Hidden layer number 2. Table II shows that 63,248 iterations were necessary to reach convergence, resulting in a training error of .0929. Table III shows that the same network structure took only 21,668 iterations to meet the convergence criteria but had a training error of .0967. The second case is the network with 15 PE's in both hidden layers. Table II indicates that 35,297 iterations were needed to reach convergence with a training error of .0747. However, Table III shows that the same network required 20,430 iterations with a training error of

Table III: NETWORK RESULTS WITH 20 INPUT PE'S AND 2 OUTPUT PE'S.

HIDDEN LAYER #1	HIDDEN LAYER #2	ITERATIONS	TRAINING ERROR	TESTING ERROR
20	0	21668	.0967	.4906
20	10	19971	.0557	.4914
20	15	23194	.0791	.4932
15	0	23533	.0756	.5007
15	5	24388	.0723	.4969
15	10	27346	.0792	.4907
15	15	20430	.0898	.4950

.0898. The other five networks in this group followed the observation that the fewer the number of iterations are needed for convergence the lower the RMS training error is. The testing files were evaluated using *RMSE.M*. The RMS testing error obtained from these files indicates that, according to the 0.1 RMS error criteria, none of these networks were trained properly to distinguish between a "whale" or "not a whale". The RMS testing error averaged around 0.5 which is considerably higher than the criteria of 0.1. These results show that, for our classification purposes, the neural networks have better performance when they are trained to distinguish among different types of whales rather than when they are trained to differentiate between "whale" or "not a whale".

Appendix C provides a scoring percentage breakdown for each network configuration. These scoring tables list the percentage of correct classifications associated with each file and overall results of the classification process.

C. PERFORMANCE OF THE SYMMETRIZED ITAKURA DISTANCE MEASURE CLASSIFICATION

The Symmetrized Itakura distance classification method bases classification on a computed distance between the observed signal and a reference signal. The mathematical derivation for this method is described in Chapter IV. The MATLAB™ code *ITAKURA.M* is configured to take an observed signal and to compute a symmetrized Itakura distance measurement with each of the five reference signals. Next, the software routine computes the distances between observed and reference signals and indicates the reference signal closest in distance to the observed signal. The number of times each reference is the closest is counted and a percentage is computed for the entire length of the observed signal. After the scoring percentage is calculated, these percentages are checked against a specified cutoff. A decision is made to classify the input signal as one of the five references or no classification is made. The distance mean is also computed for each comparison. If a reference distance is above the specified cutoff percentage but, its distance mean is above a set threshold, no classification is

made. Table IV records the Symmetrized Itakura Distance classification results for a cutoff percentage score of 51.00% and a threshold of 2.00.

All 16 biological data files were used in conjunction with the *ITAKURA.M* code to test the classification application of the Symmetrized Itakura Distance measure. A percentage score is provided for each reference to the input signal. This percentage represents the portion of the total length of the input signal where the reference and the input signal had the lowest distance measurement. The ID RESULT gives the resulting classification from the above mentioned criteria. This result specifies one of the five references; Gray, Humpback, Killer, Sealions (SLIONS) or Sperm Whales, None(as in no classification or Wrong). A wrong result indicates that the classification reached did not match the input signal even though all the necessary classification criteria were met.

The AORCA512, ABERD512, and ABRD2512 input files resulted in erroneous classifications. The AORCA512 file had 84.54% of its lowest distance computations with the Sperm Whale reference (ISREF) and its distance computational mean was below the threshold. However, note that this is a wrong classification as AORCA512 is the NOSC data file for a Killer Whale and not a Sperm Whale. It is quite coincidental that this file classified with the Sperm Whale reference (ISREF) which is from the same source. Furthermore, the entire ASP512 file data was correctly classified against the Sperm Whale

Table IV: SYMMETRIZED ITAKURA DISTANCE CLASSIFICATION

INPUT FILE	IGREF % SCORE	IHREF % SCORE	IKREF % SCORE	ILREF % SCORE	ISREF % SCORE	ID RESULT
AGRAY512	94.74	0	0	3.16	2.11	GRAY
AHUMP512	9.56	16.91	34.56	11.03	27.94	NONE
AKILL512	6.03	0	43.10	35.34	15.52	NONE
ALION512	0	0	0	51.56	48.44	SLION
AORCA512	1.03	0	0	14.43	84.54	WRONG
ABERD512	71.59	16.48	3.41	8.52	0	WRONG
ABRD2512	80.83	3.33	10.83	5.00	0	WRONG
ASP512	1.03	0	0	0	98.97	SPERM
ABIOA512	10.27	73.97	15.75	0	0	NONE
ABIOB512	8.22	69.86	21.92	0	0	NONE
ABIOC512	6.85	92.47	0.68	0	0	NONE
AB2A512	22.60	71.92	5.48	0	0	NONE
AB2B512	15.07	78.08	6.85	0	0	NONE
AB2C512	10.46	77.12	12.42	0	0	NONE
AICE512	98.48	0.67	0.84	0	0	NONE
AFLIP512	12.65	61.72	25.63	0	0	NONE

reference. Both of the Bearded Seal files, ABERD512 and ABRD512, classified as Gray Whales. Again, the Gray Whale reference (IGREF) percentage score was the largest of all the references and had its distance computational mean below the threshold. We would have expected the results to show scoring percentages spread across all five references, similar to the results of AHUMP512, or its mean well above the threshold.

The result of the AHUMP512 classification shows a mix of percentage scores for the Killer Whale reference (IKREF) with

the largest percentage of the five but below the cutoff. The segment of sound that was chosen to represent the Humpback Whale must not have been the best representation for the entire signal. The results show that there was no real dominant comparison available to make a decision.

The AKILL512 classification does show an accurate classification even though the ID RESULT specifies otherwise. The IKREF percentage of 43.10 is the largest percentage of the five but does not meet the minimum cutoff percentage. The Sealion reference (ILREF) had the next closest percentage with 35.34.

All of the ABIO's and AB2's had results of no classification even though every file had its largest reference percentage score with IHREF. These six files from NUSC with unknown origins had distance computational means far above the established threshold. This is evident in the graphical representation of the Symmetrized Itakura distance classifications for BIOB512 and AB2C512 found at the end of this chapter. The AICE512 and AFILP512 are also signals from NUSC. The classification for AICE512 shows an overwhelming comparison to IGREF even though this is a file of ice cracking. The result is no classification due to the mean of the reference distance computational exceeding the established threshold. AFLIP512 is a data file modeled from a porpoise whistle also from NUSC. The classification results specify no classification which is correct but for the wrong reasons.

IHREF had the largest reference percentage score but the mean of this reference distance computation is again above the threshold.

The three classifications that were correctly done based on the reference percentage score were AGRAY512, ALION512, and ASP512. Both AGRAY512 and ASP512 had unquestionably the majority of their reference percentage scores in the appropriate categories. ALION512 was a correct classification based strictly on its reference percentage score exceeding the threshold. This classification was extremely close to one of a "no-classification" with an almost tie with ISREF.

Graphical representations of the Symmetrized Itakura Distance classifications are available in Figures 26 through 35. Only one of the six results of the NUSC Sperm Whales is presented due to their similarities. All graphs can be read the same way. The solid line represents the Gray Whale reference distance measurement, the dashed line represents the Humpback Whale reference distance measurement, the dotted line represents the Killer Whale reference distance measurement, the dash-dotted line represents the Sealion reference distance measurement, and the asterisks represent the Sperm Whale reference distance measurement.

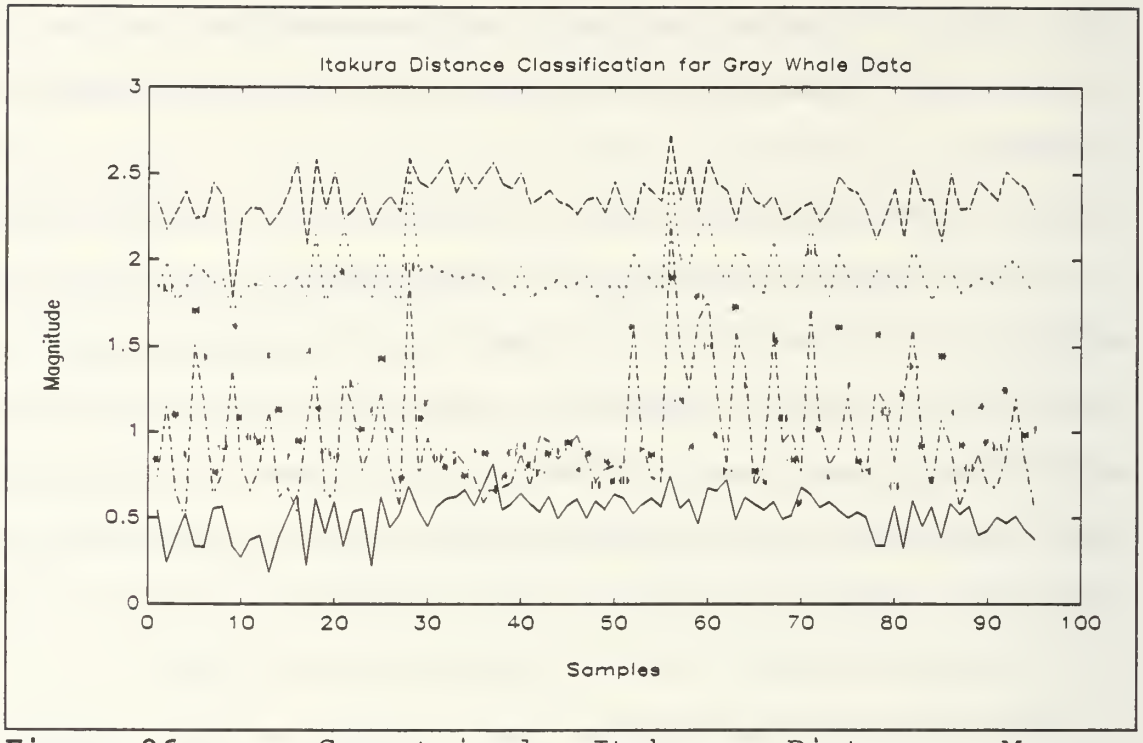

Figure 26: Symmetrized Itakura Distance Measure Classification for Gray Whale.

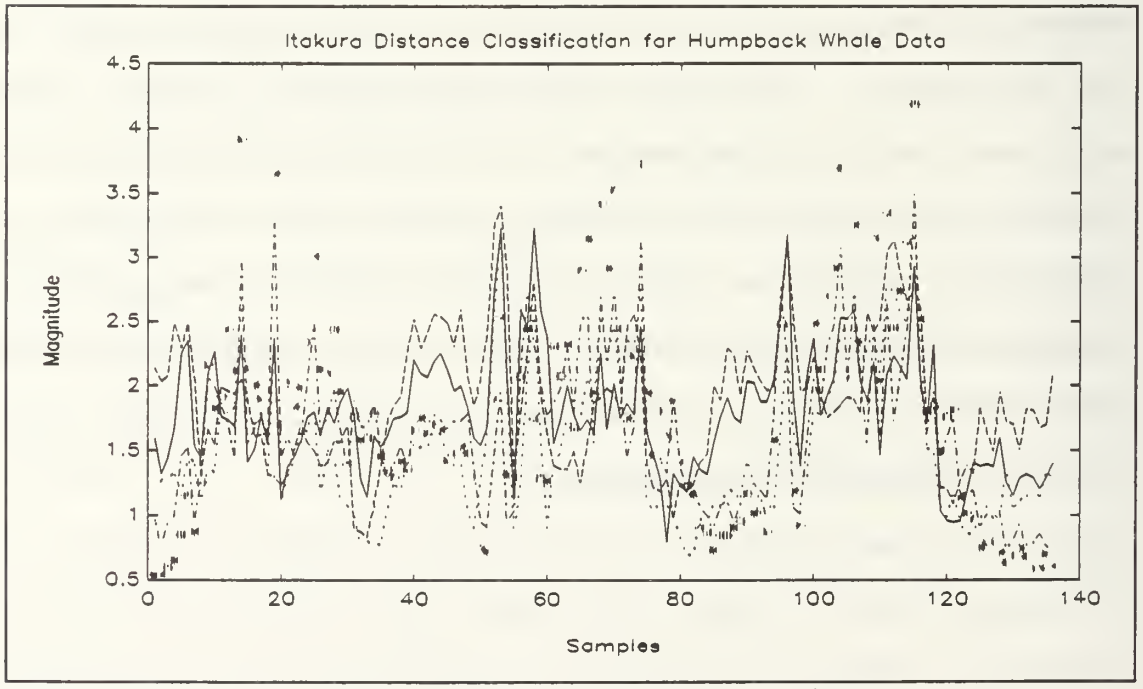

Figure 27: Symmetrized Itakura Distance Measure Classification for Humpback Whale Data.

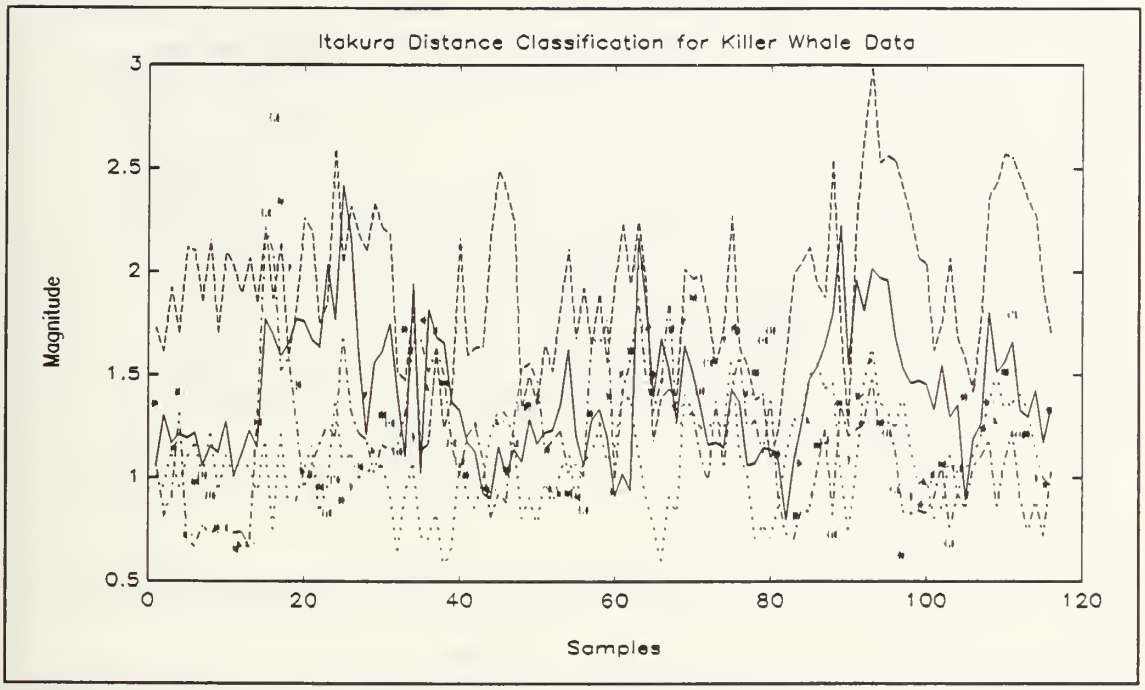

Figure 28: Symmetrized Itakura Distance Measure Classification for Killer Whale Data.

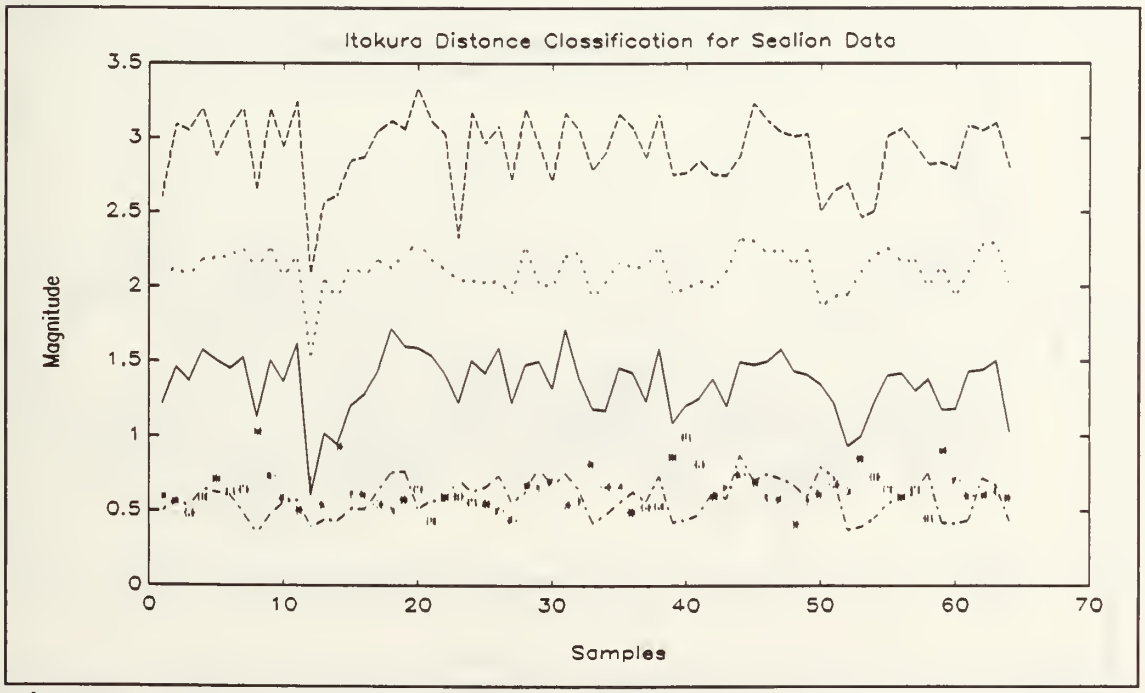

Figure 29: Symmetrized Itakura Distance Measure Classification for California Sealion Data.

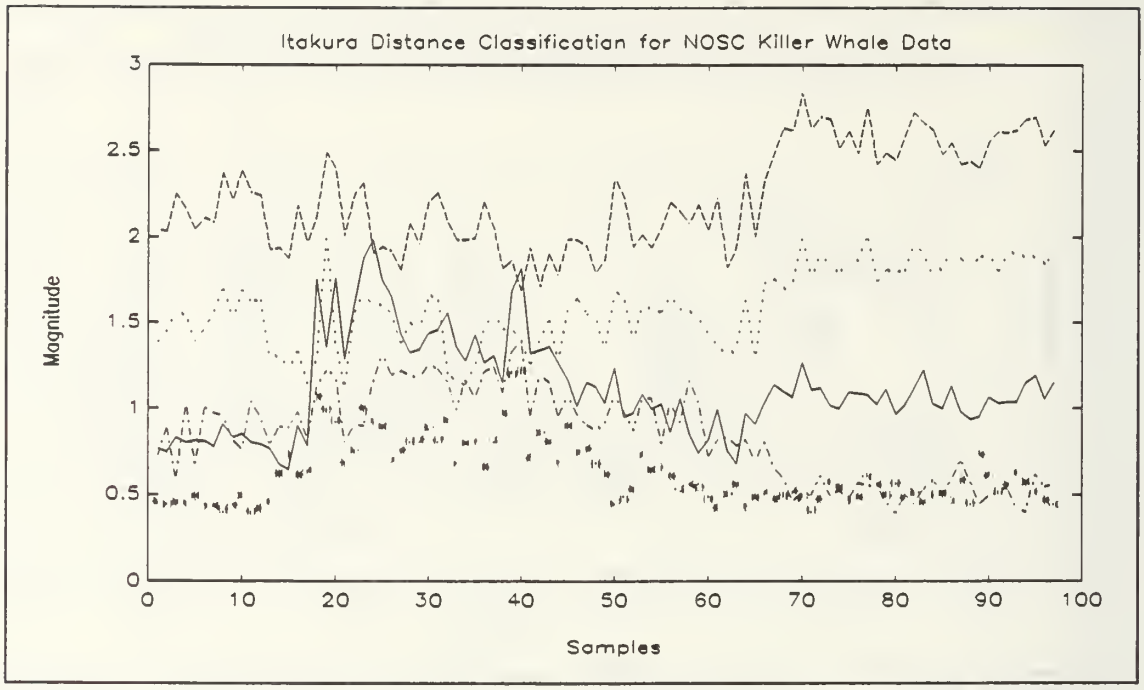

Figure 30: Symmetrized Itakura Distance Measure Classification for NOSC Killer Whale Data.

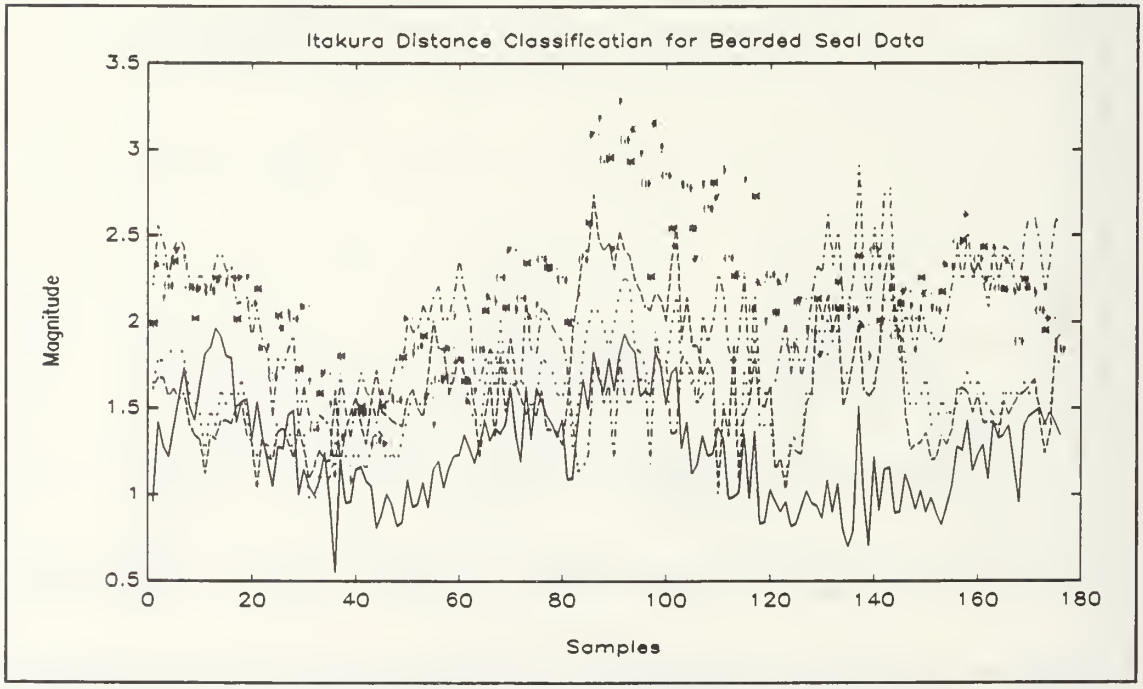

Figure 31: Symmetrized Itakura Distance Measure Classification for Bearded Seal Data.

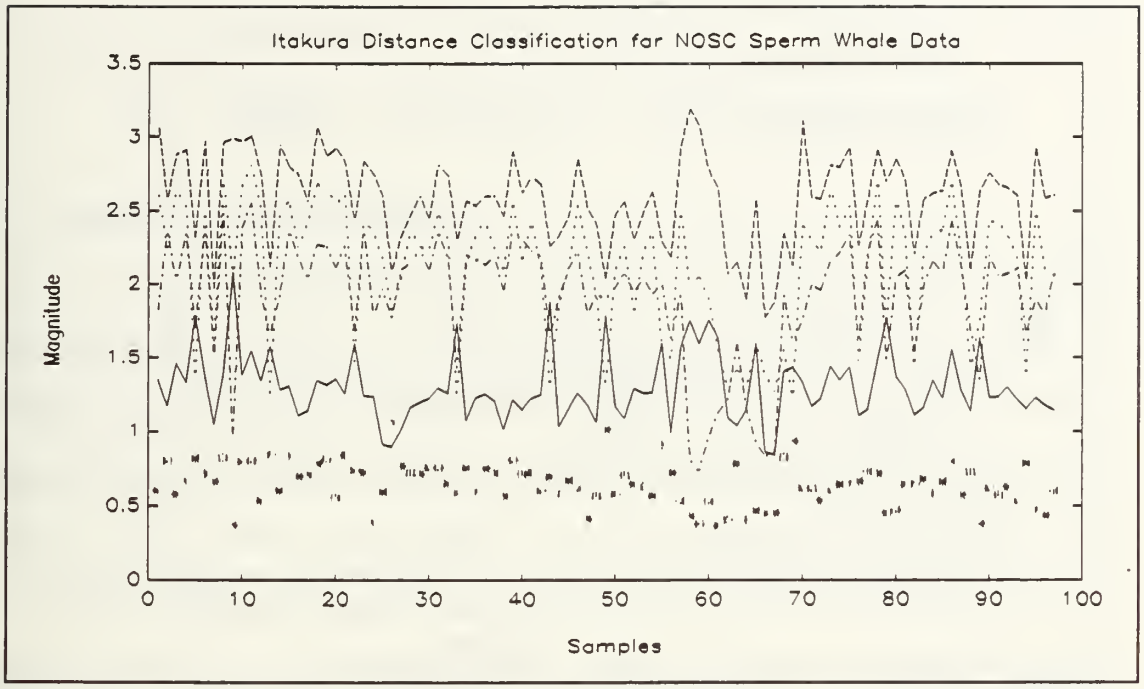

Figure 32: Symmetrized Itakura Distance Measure Classification for NOSC Sperm Whale Data.

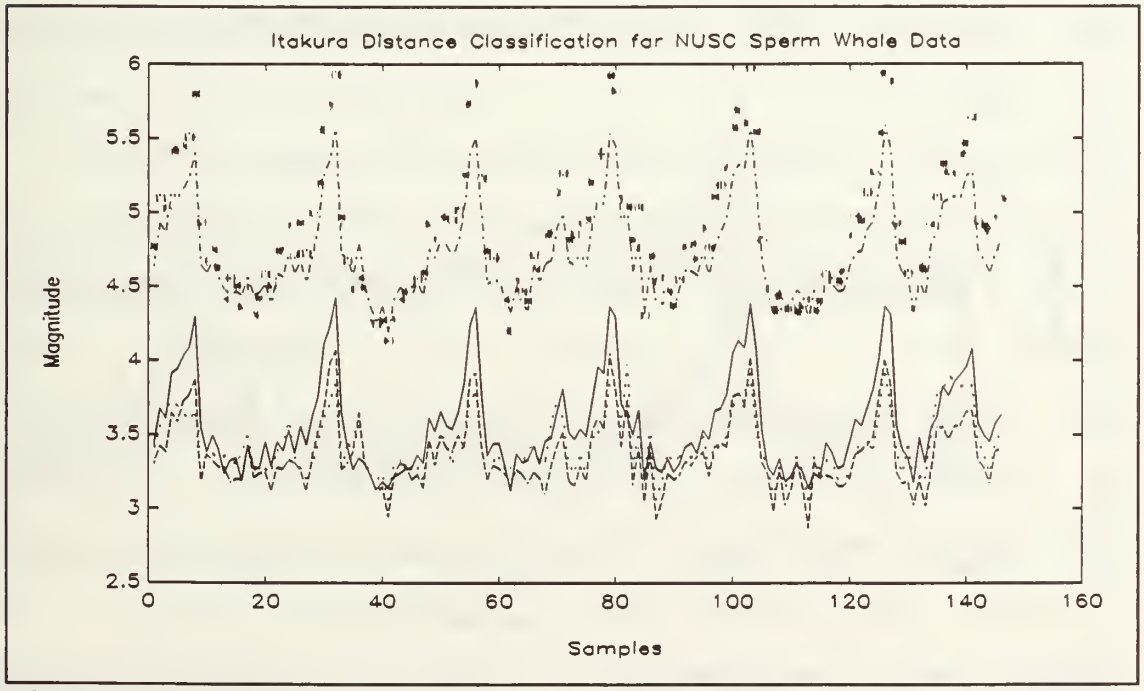

Figure 33: Symmetrized Itakura Distance Measure Classification for NUSC Sperm Whale Data.

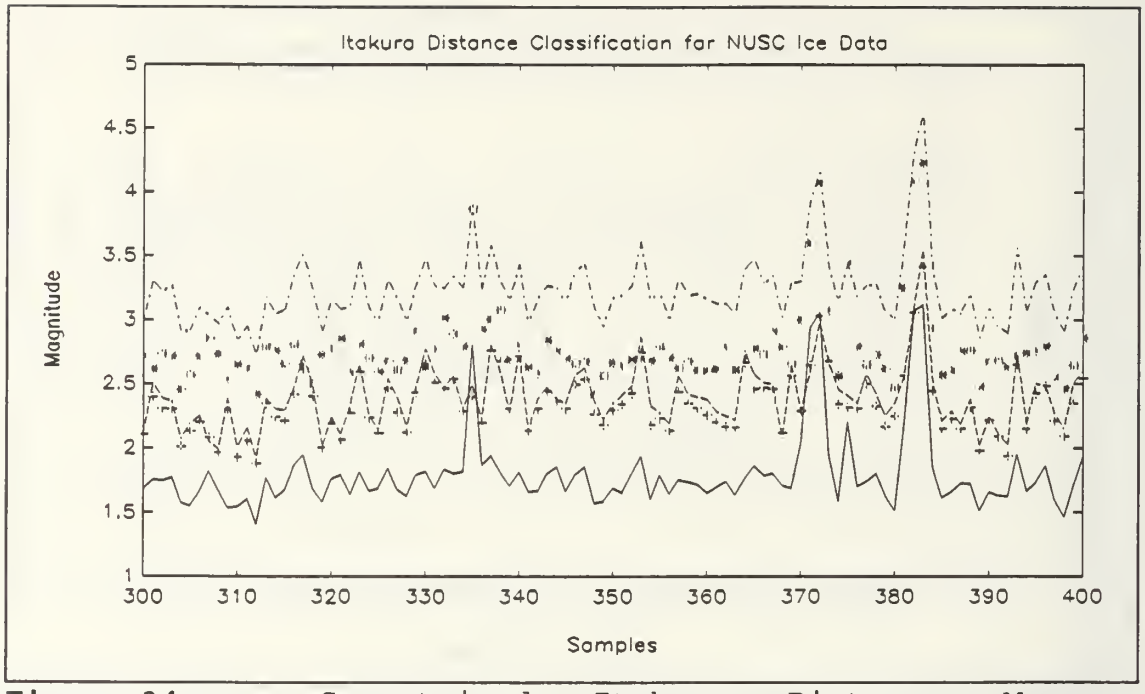

Figure 34: Symmetrized Itakura Distance Measure Classification for NUSC Ice Data.

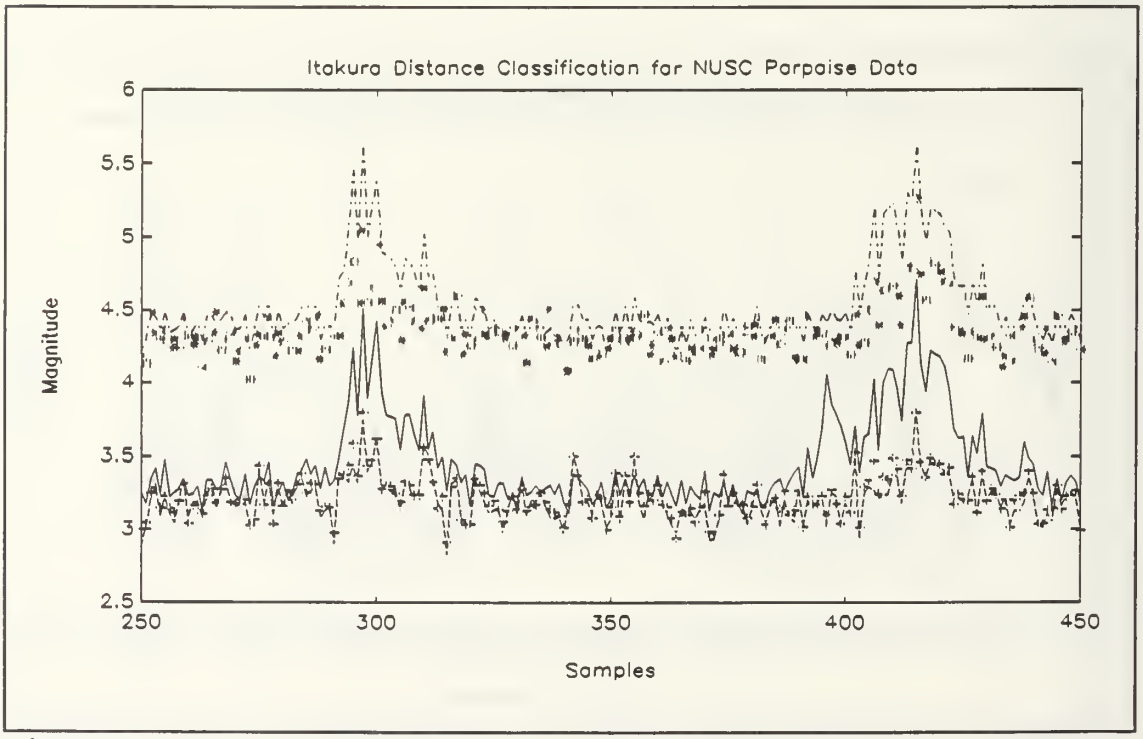

Figure 35: Symmetrized Itakura Distance Measure Classification for NUSC Porpoise Data.

VI. SUMMARY, CONCLUSION, AND RECOMMENDATIONS

A. SUMMARY AND CONCLUSION

In this thesis we have studied AR techniques to model time-domain signals. The AR algorithm is based on Prony's method for modeling transient signals as the sum of complex, damped exponentials. Next, SVD enhancement was introduced to reduce the effect of noise in the modeling procedure. The Prony-SVD method appears to work well in detecting and modeling impulsive and periodic signals even in relatively low SNR, considering the model was consistent for all data types. The choice of model number and useable singular values in the algorithm was accomplished visually. Methods for order selection can be incorporated to make this processes more systematic [Ref. 8:p. 389].

The back-propagation neural network used to classify four species of whale from the model parameters proved itself a viable classification method. The attempt to build a neural network to identify between a "whale" or "not a whale" was not a successful classification method using the same model parameters. The data presented to train the network was different enough so that the network required more information to be trained successfully. The results from Chapter V demonstrate that the amount of training data has a direct

effect on network performance. Even though adding a hidden layer resulted in a faster convergence and fewer iterations during the training phase, the network was not necessarily better trained. Using the data that was available, all species were successfully classified with the exception of the *AORCA512*, *ABERD512* and *ABRD2512*. In every network configuration, *AORCA512* was classified as "other". This result appears to be more a factor of the quality of the data than the network performance. The only other factor that might have been a factor concerning this specific piece of data is that it has a different sampling rate. The sperm whale data from the same source, however, did not present the same problem.

The Symmetrized Itakura Distance Measure classification method provides some interesting classification possibilities. The quality of the signal references determines the reliability of the Symmetrized Itakura Distance. Overall, the SI distance classification method did not do as well as the neural network. This could be both a result of the limited data available and the choice of the reference signals. For example, the humpback whale did not classify correctly. Note that the humpback whale sound has a larger dynamic range than the other signals. These dynamic characteristics appear to have a significant effect on the spectral ratio.

In conclusion, the neural network provide a better classification mechanism using AR model parameters than the

Symmetrized Itakura distance measurement does for the data available in this thesis. Results of both classification methods are summarized in Table V.

Table V: SUMMARIZATION OF CLASSIFICATION METHODS.

FILE NAME	NEURAL NETWORK	SYMMETRIZED ITAKURA
AGRAY512	Classified Correctly	Classified Correctly
AHUMP512	Classified Correctly	No Classification
AKILL512	Classified Correctly	No Classification
ALION512	Classified Correctly	Classified Correctly
AORCA512	Wrong Classification	Wrong Classification
ABERD512	Classified Correctly	Wrong Classification
ABRD2512	Classified Correctly	Wrong Classification
ASP512	Classified Correctly	Classified Correctly
ABIOA512	Classified Correctly	No Classification
ABIOB512	Classified Correctly	No Classification
ABIOC512	Classified Correctly	No Classification
AB2A512	Classified Correctly	No Classification
AB2B512	Classified Correctly	No Classification
AB2C512	Classified Correctly	No Classification
AICE512	Classified Correctly	No Classification
AFLIP512	Classified Correctly	No Classification

B. RECOMMENDATIONS

Additional work on this thesis is recommended. A method to model the data signal recursively instead of segmenting it as was accomplished in this thesis is recommended. In addition, the SVD technique can be enhanced by incorporating

a self-adjusting method to select either the model order or the usable number of singular values.

This thesis can be further improved by expanding the data base to provide a more comprehensive set of biological signals. This should include transient acoustic signals to enhance classification of biologics. Furthermore, a reference library of biological signals that truly characterize the biologics to be classified for use in the Symmetrized distance needs to be established. Finally, methods to determine the minimum size of the neural network architecture required to perform the biological classification task are recommended.

APPENDIX A - SOFTWARE AND HARDWARE DESCRIPTION

The thesis computer work was done using the two commercially available software packages described below.

PRO-MATLAB
for Sun Workstations
Version 3.5i
The Mathworks, Inc.
Cochituate Place
24 Prime Park Way
Natwick, MA 01760
Phone: (508)653-1415

A short description is provided from the MATLAB User's guide:

> MATLAB is a high-performance interactive software package for scientific and engineering numeric computations. MATLAB integrates numerical analysis, matrix computation, signal processing, and graphics in an easy-to-use environment where problems and solutions are expressed just as they are written mathematically - without traditional programming.

NeuralWorks Professional II/Plus
NeuralWare, Inc.
Penn Center West, Building IV, Suite 227
Pittsburgh, PA 15276
Phone: (412)787-8222

A short description is provided from the NeuralWare User's guide:

> NWorks is the core of a collection of products, that taken together, provide the most complete and up-to-date neural network development and deployment environment available. It is a multi-paradigm neural network proto-typing and development tool. It can be used to design, build, train, test and deploy neural networks to solve complex real-world problems. Dozens of well-known built-in network types can be quickly generated or design your own custom networks. Numerous enhancements to backpropagation and

other networks are included. Networks are displayed graphically, PEs and connections can be easily added or deleted. Dozens of math functions and learning rules can be mixed and matched. Typical applications include: financial analysis, signal processing, targeted marketing, robotics and automation, pattern recognition, and optimization.

The computers used in the thesis work were a Sun SPARC station 1+ with 16 megabytes (MB) or random access memory (RAM) and a more than 1 giga-bytes of hard disk storage available through ECE server network and a Gateway 2000 IBM/PC compatible 386-25 MHz with math-coprocessor, 4 MB of RAM and a total of 280 MB or hard disk storage.

APPENDIX B - COMPUTER CODE

This appendix presents the MATLAB™ code used in the thesis. The format of the appendix presents the MATLAB™ program code. This code was run from inside the MATLAB™ workspace as either a mfile or a function.

A. ITAKURA.M

```
%  itakura.m
%
%  This routine will take the fft of a modeled biological
%  signal and then classify the signal in regards to a
%  computed Itakura distance measurement from five reference
%  signals.
%

clear

cutoff = .51;
threshold = 2.00;

%  Load the Itakura reference signals
load IGref
load IHref
load IKref
load ILref
load ISref

names = ['Gray Whale    '
         'Humpback Whale'
         'Killer Whale  '
         'Sealions      '
         'Sperm Whale   '];

%  Load the file wish to classify
load aflip512;
[a,b]=size(aflip512);

%  Calculate the fft on biodata AR parameters
```

```
for k=1:b
  q=[1, -aflip512(:,k)'];
  g=fft([q,zeros(1,512-length(q))]);
  gfft = 1 ./ abs(g);
  biocomp=abs(gfft) .* abs(gfft);

  x1 = IGref ./ biocomp;
  x2 = biocomp ./ IGref;

  x3 = IHref ./ biocomp;
  x4 = biocomp ./ IHref;

  x5 = IKref ./ biocomp;
  x6 = biocomp ./ IKref;

  x7 = ILref ./ biocomp;
  x8 = biocomp ./ ILref;

  x9 = ISref  ./ biocomp;
  x10 = biocomp ./ ISref;

   alpha = 1/(512 ^2);

%  Do Itakura Distance calculation

  dcomp(:,k) = [0.5*(log(alpha*sum(x1)*sum(x2)))
                0.5*(log(alpha*sum(x3)*sum(x4)))
                0.5*(log(alpha*sum(x5)*sum(x6)))
                0.5*(log(alpha*sum(x7)*sum(x8)))
                0.5*(log(alpha*sum(x9)*sum(x10)))];

  [m,i(k)] = min(dcomp(:,k));

end;

%  Classification of inputed biological data

num = [length(find(i==1))
       length(find(i==2))
       length(find(i==3))
       length(find(i==4))
       length(find(i==5))] / b   %  Results in a fractional
breakdown

%    Classification rule: if max value < cutoff -- no
classification

[maximum,imax] = max(num);
   if maximum < cutoff
      disp('No Classification');
      disp('All values below established cutoff');
```

```
   else
      meanimax = mean(dcomp(:,imax));
      if meanimax > threshold
         disp('No Classification');
         disp('This data is not one of the four whales or a
sealion');
      else
         disp('The Data is classified as:');
         disp(names(imax,:));
      end;
    end;
pause

% Plot the distance results
t = 1:b;
plot(t,dcomp(1,:),'-',t,dcomp(2,:),'--',t,dcomp(3,:),':',t,...
dcomp(4,:),'-.',t,dcomp(5,:),'*')


title('Itakura Distance Classification for a Porpoise')
```

B. ITAREF.M

```
%  itaref.m
%
%  This routine will take the segmented parts of modeled
%  biologic sounds of and use these as the references for the
%  Itakura Distance reference to classify biological data.
%
clear

%  Load the appropriate files needed
load humparp
load lionarp
load spermarp
load killarp
load grayarp

%  Create the Humpback Whale Sound segment Itakura Reference
   h = [1,-humparp'];
   hp= fft([h,zeros(1,512-length(h))]);
   fhump = 1 ./ abs(hp);
   IHref = abs(fhump) .* abs(fhump);
plot(fhump');title('FFT of Humpback Whale Itakura Reference')
keyboard

save IHref IHref
```

```
%    Create the Sealion reference for Itakura Distance
Classification
   q = [1, -lionarp(:,1)'];
   g = fft([q,zeros(1,512-length(q))]);
   flion = 1 ./ abs(g);
plot(flion','wg')
title('FFT of Sealion Sound AR Parameters to Form Itakura
Reference')

keyboard

ILref = abs(flion) .* abs(flion);
save ILref ILref

%  Create the Sprem Whale Sound Itakura Reference
   sp = [1, -spermarp(:,1)'];
   s=fft([sp,zeros(1,512-length(sp))]);
   fsperm = 1 ./ abs(s);
   ISref = abs(fsperm) .* abs(fsperm);
plot(fsperm');title('FFT of Sperm Whale Itakura Reference')
keyboard
save ISref ISref

%  Create the Killer Whale sound Itakura Reference
   l = [1, -killarp'];
   p = fft([l,zeros(1,512-length(l))]);
   fkill = 1 ./ abs(p);
   IKref = abs(fkill) .* abs(fkill);

plot(fkill');title('FFT of Killer Whale Itakura Reference')
keyboard
save IKref IKref


%  Create the Gray Whale Sound Itakura Reference
 l2=[1,-grayarp'];
 p2=fft([l2,zeros(1,512-length(l2))]);
 fgray = 1 ./ abs(p2);
 IGref =abs(fgray) .* abs(fgray);

plot(fgray');title('FFT of Gray Whale Itakura Reference')
keyboard

save IGref IGref
```

C. PRONYSVD.M

```
function [a_hat,err]=pronysvd(y,n)
% function a_hat=pronysvd(y,n)
% compute the AR coefficients of data points y
% n=max. order desired
%
%  This file combines prony and svd methods to model a segment
%  of biological data.   The segment of data (y) is provided
%  along with the model order(n).  The model parameters
%  (a_hat) are computed and an error measurement (err) between
%  the original segment and modeled signal is provided.
%
ny=size(y);
if ny(2)>ny(1), y=y'; end;
y=y-mean(y);
N=length(y);
%  Compute the correlation matrix of y(l)
sum=0;
for t=n+1:N;
   sum=sum+y(t-1:-1:t-n,1)*y(t-1:-1:t-n,1)';
end;

Bias=1/(N-n);
Ryy=Bias*sum;

%  Compute the SVD of Ryy

[u,s,v]=svd(Ryy);

%  Plot the singular values of s
clg
plot(diag(s),'wg')
pause
m=input('enter number of s_values to be saved » ');
% first: subtract min. s_value:
Rxx=Ryy-s(n,n)*eye(n);
%
%second: compute pseudoinv. of Rxx keeping only m s_values:
ds=diag(s);
dsi(1:m)= 1 ./ ds(1:m); dsi(m+1:n)=zeros(1,n-m);
si=diag(dsi);

%  Compute H'x in terms of y
sum2=0;
for t=n+1:N;
  sum2=sum2+y(t-1:-1:t-n,1)*y(t,1);
end;
p=Bias*sum2;

%  To solve for a parameters first must solve for Rxx
```

```
%  since Rxx is not invertible(not full rank) use the
%  pseudoinverse.  Must look at the s matrix from
%  above and set values close to zero to zero(<1) the
%  rest of the values will be set inverted.
Rxx_inv = u*si*u';

%  Compute the a estimated parameters from Rxx'*p
a_hat = Rxx_inv*p;

% Compute estimated spectrum
q=[1, -a_hat'];
g=fft([q,zeros(1,N-n-1)]); fhat=1 ./ abs(g);
Y=abs(fft(y));
fhat=fhat/sqrt(fhat*fhat'); Y=Y/sqrt(Y'*Y);

%  Plot the estimated against the original signal
plot([fhat',Y],'wg')

%  Compute the error between the estimated and original
diff=Y-fhat';
D=(diff)'*diff;
err=(sqrt(D))/N;
```

D. RESULT3.M

```
function [zz] = result3(tst,start,finish)
%
%  This function scores the 11 different files used
%  to make up the B512TST.NNA for neural network testing
%  and provides a percentage.
%         tst == file to score
%         start == starting position in the file
%         finish == ending position in the file

x = tst(start:finish,6:10);
[a,b] = size(x);

z = [];
for i = 1:a
   Xmax = max(x(i,:));
   xx = x(i,:) == Xmax;
   z = [z; xx];
end
zz = sum(z)/a;
```

E. RESULTS.M

```
%  RESULTS.M
%
%   This m-file process percentage results
%   from neural network testing

clear
clg
load tst47.nnr
tst = tst47;
zz1=result3(tst,1,32);
zz2=result3(tst,33,77);
zz3=result3(tst,78,116);
zz4=result3(tst,117,137);
zz5=result3(tst,138,169);
zz6=result3(tst,170,201);
zz7=result3(tst,202,354);
zz8=result3(tst,355,500);
zz9=result3(tst,501,620);
zz10=result3(tst,621,820);
zz11=result3(tst,821,1020);

z=[zz1;zz2;zz3;zz4;zz5;zz6;zz7;zz8;zz9;zz10;zz11];

zcent = 100*z
```

F. RMSE.M

```
% RMSE.M
%
%   This m-file takes test file results from a neural
%   network in a *.nnr format and computes the RMS error
%   for the entire file.  Row speciefies the length of the
%   file and col is the number of output processing elements
%   in the neural network.

clear
clg
row = 1020;
col = 5;
pts = row * col;

load tst46.nnr
data = tst46;
sum = 0;
for i = 1: row
```

```
  for j = 1: col
    x(j) = (data(i,j) - data(i,j+5)) ^ 2;
    sum = sum + x(j);
  end
 end

rms_error = sum/pts
```

G. RUNDATA.M

```
%  rundata.m
%
%  This m-file is used to manipulate a large
%  biological file.  Uses function a_hat=pronysvd(y,n)

load humpback           % Load the biological data file to
segment

err=zeros(1,68);
y=zeros(1024,68);
Ahat=zeros(20,68);
k=0;
for t=1:68;
  y1=humpback(1+k:1024+k,1);
  k=1024 + k;
  y(:,t)=y1;
 [a_hat,err]=pronysvd(y1,20);    %  Computes  the  AR  model
parameters   Ahat(:,t)=a_hat;
  error(1,t)=err;

end;
```

APPENDIX C - NEURAL NETWORK SCORING PERCENTAGES

This appendix provides a tabular listing of the scoring percentages of each neural network configuration. Each table lists by category the percentage of correct classifications for every type presented to the neural network. The name of the neural network is given by a set of four numbers separated by dashes, i.e. 20/15/5/5. The name corresponds to 20 input processing elements, 15 processing elements in hidden layer 1, 5 processing elements in hidden layer 2 and, 5 processing elements in the output layer.

SCORING PERCENTAGES FOR NEURAL NETWORK 20/15/0/5

FILE	GRAY	HUMPBACK	KILLER	SPERM	OTHER
AGRAY512	100	0	0	0	0
AHUMP512	0	80	13.33	0	6.67
AKILL512	0	28.21	69.23	0	2.56
ALION512	0	0	0	0	100
AORCA512	0	0	3.13	3.13	93.75
ASP512	0	0	3.13	81.25	15.62
AB2C512	0	0	0	73.86	26.14
ABIOA512	0	0	0	71.23	28.76
ABRD2512	0	0	4.17	.83	95.0
AICE512	0	0	0	0	100
AFLIP512	0	0	0	7.0	93.0

SCORING PERCENTAGES FOR NEURAL NETWORK 20/15/5/5

FILE	GRAY	HUMPBACK	KILLER	SPERM	OTHER
AGRAY512	100	0	0	0	0
AHUMP512	0	80.0	6.67	0	13.33
AKILL512	0	20.51	61.54	0	17.95
ALION512	0	0	0	0	100
AORCA512	0	0	3.13	0	96.87
ASP512	0	0	3.13	84.37	12.5
AB2C512	0	0	0	94.12	5.88
ABIOA512	0	0	0	93.84	6.16
ABRD2512	0	0	16.67	.83	85.0
AICE512	0	0	0	0	100
AFLIP512	0	0	0	34.5	65.5

SCORING PERCENTAGES FOR NEURAL NETWORK 20/15/10/5

FILE	GRAY	HUMPBACK	KILLER	SPERM	OTHER
AGRAY512	100	0	0	0	0
AHUMP512	0	66.67	15.56	0	17.77
AKILL512	0	25.64	69.23	0	5.13
ALION512	0	0	0	0	100
AORCA512	0	0	0	3.12	96.88
ASP512	0	3.12	3.12	84.38	9.38
AB2C512	0	0	.65	93.46	5.88
ABIOA512	0	0	0	92.47	7.53
ABRD2512	0	0	8.33	.83	90.83
AICE512	0	0	0	0	100
AFLIP512	0	0	0	38.0	62.0

SCORING PERCENTAGES FOR NEURAL NETWORK 20/15/15/5

FILE	GRAY	HUMPBACK	KILLER	SPERM	OTHER
AGRAY512	100	0	0	0	0
AHUMP512	0	73.33	17.78	0	8.89
AKILL512	0	20.51	76.92	0	2.57
ALION512	0	0	0	0	100
AORCA512	0	0	6.25	9.37	84.38
ASP512	0	0	3.12	93.75	3.13
AB2C512	0	0	.65	84.97	14.37
ABIOA512	0	0	0	81.51	18.49
ABRD2512	0	0	9.17	0	90.83
AICE512	0	0	0	0	100
AFLIP512	0	0	0	24.5	75.5

SCORING PERCENTAGES FOR NEURAL NETWORK 20/20/0/5

FILE	GRAY	HUMPBACK	KILLER	SPERM	OTHER
AGRAY512	100	0	0	0	0
AHUMP512	0	80	15.56	0	0
AKILL512	0	30.77	66.67	0	2.56
ALION512	0	0	0	0	100
AORCA512	0	0	18.75	9.38	71.85
ASP512	0	3.12	3.12	90.63	3.12
AB2C512	0	0	.65	92.81	6.54
ABIOA512	0	0	0	95.21	4.79
ABRD2512	0	0	6.67	0	93.33
AICE512	0	0	0	0	100
AFLIP512	0	0	0	30	70.0

SCORING PERCENTAGES FOR NEURAL NETWORK 20/20/10/5

FILE	GRAY	HUMPBACK	KILLER	SPERM	OTHER
AGRAY512	100	0	0	0	0
AHUMP512	0	84.44	6.67	0	8.89
AKILL512	0	46.15	46.15	0	7.69
ALION512	0	0	0	0	100
AORCA512	0	0	15.63	6.25	78.12
ASP512	0	12.5	0	71.88	15.62
AB2C512	0	0	.65	32.03	67.32
ABIOA512	0	0	0	19.86	80.14
ABRD2512	0	0	7.5	0	92.5
AICE512	0	0	0	0	100
AFLIP512	0	0	0	1.0	99.0

SCORING PERCENTAGES FOR NEURAL NETWORK 20/20/15/5

FILE	GRAY	HUMPBACK	KILLER	SPERM	OTHER
AGRAY512	100	0	0	0	0
AHUMP512	0	77.78	11.11	0	11.11
AKILL512	0	38.46	46.15	0	15.38
ALION512	0	0	0	0	100
AORCA512	0	0	9.37	0	90.63
ASP512	0	3.12	3.12	84.38	9.38
AB2C512	0	0	.65	84.31	15.03
ABIOA512	0	0	0	79.45	20.55
ABRD2512	0	0	8.33	.833	90.83
AICE512	0	0	0	0	100
AFLIP512	0	0	0	21.0	79.0

REFERENCES

1. S. L. Marple, *Digital Spectral Analysis with Applications*, Prentice-Hall, Englewood, N.J., 1987.

2. S. K. Kay, *Modern Spectral Estimation: Theory and Application*, Prentice-Hall, Englewood, N.J., 1988.

3. E. W. Bingham and C. H. Velasco, "Application of Human Speech Processing Techniques on Marine Life," (private communications) Speech Signal Processing EC 4410, 8 June 1992.

4. S. L. Marple and T. Brotherton, "Detection and Classification of Short Duration Underwater Acoustic Signals by Prony's Method," *IEEE TranAcous., Speech, Sig. Proc.*, Vol. CH-2977, pp. 1309-1312, July 1991.

5. R. Hecht-Nielsen, *Neurocomputing*, Addison-Wesley Publishing Company, 1990.

6. NeuralWare, Inc., *NeuralWorks Professional II/PLUS and NeuralWorks Explorer Reference Manuals*, Pittsburgh, PA., 1991.

7. K. Lashkari, J. Abel, and B. Mcquiston , Naval Underwater Systems Center, Report 5525-01, *Classification of Underwater Transients*, September 1986.

8. M. Wax and T. Kailath, "Detection Of Signals by Information Theoretic Criteria," *IEEE TranAcous., Speech, Sig. Proc.*, Vol. ASSP-33, NO. 2, pp. 387-392, April 1985.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Prof. Roberto Cristi, Code EC/CX Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
5.	Prof. Monique P. Fargues, Code EC/FA Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	2
6.	LT Martha M. VanDerKamp United States Naval Academy Department of Electrical Engineering - Stop 14B Annapolis, MD 21402-1220	1
7.	Nicole Crane Hopkins Marine Station of Standford University Pacific Grove, CA 93950	1
8.	Dr. Khosron Lashkari 160 Central Ave. Pacific Grove, CA 93950	1

Thesis
V16152 VanDerKamp
c.1 Modeling and classification of biological signals.

Thesis
V16152 VanDerKamp
c.1 Modeling and classification of biological signals.

AMERICAN BINDERY
1
10-93

DUDLEY KNOX LIBRARY
3 2768 00033425 4